

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Departamento de Ingeniería Informática



TRABAJO DE FIN DE GRADO:

GESTIÓN DE RUTAS Y TOMA DE DECISIONES EN EL ENTORNO DE SIMULACIÓN STI-SIM

Autor: Zamora España, Víctor Manuel

Tutores: Sanchis de Miguel, Araceli
Ledezma Espino, Agapito

Junio 2016

“No siempre se consigue lo que se quiere,
pero si se intenta, se consigue lo que se necesita.”
- *Mick Jagger, Keith Richards.*

Agradecimientos

Este es el broche de oro que cierra los cuatro años más intensos de mi vida. Es por eso que conviene mencionar a todas aquellas personas que me han apoyado tanto en los buenos momentos como en los malos.

En primer lugar, mis padres Luis y M^a José, y a mis hermanos, Luis Alberto y José Carlos, que son los que más han tenido que aguantar. Todos ellos me han educado para ser quien soy y para llegar donde he llegado. Estoy infinitamente agradecido por haber tenido tanta suerte.

Muchas gracias también a mis tutores los profesores Araceli y Agapito por, además de despertar mi interés en el campo de la Inteligencia Artificial con sus clases, darme la oportunidad de trabajar con ellos y por guiar este proyecto desde el principio hasta el final.

Gracias a mis compañeros de laboratorio por su acogida y por su maravillosa convivencia, en especial a Óscar, por solucionar todas y cada una de las dudas que me han surgido día a día y por discutir conmigo las ideas y posibles soluciones a los problemas.

También agradecer a los que han sido mis compañeros en la universidad, en las clases, las prácticas, las horas de estudio... Poco a poco al final siempre lo sacamos todo. Merecen especial mención Saúl y Alejandro, que desde el principio hemos permanecido juntos, y Álvaro, que aunque ha aparecido hacia el final de estos cuatro años, ha sido un gran apoyo y una gran ayuda aportando otros enfoques cuando me quedaba atascado.

Por último, también mencionar a mis amigos de siempre, Carlos, Del Prado y Lidia, y a mi persona especial, Celia, que siempre han estado ahí para apoyarme en los momentos en los que mi moral estaba más baja y cuando dudaba de mí mismo. Han sido el empujón hacia delante en los momentos más cruciales.

Resumen

La seguridad en las carreteras es un tema que tiene gran importancia a nivel mundial. El porcentaje de población que muere cada año por causas relacionadas ha ido creciendo en los últimos años. Según el *Informe sobre la situación mundial de la seguridad vial 2015* de la *Organización Mundial de la Salud (OMS)* [1], 1,25 millones de personas mueren cada año por accidentes de tráfico, sin contar los accidentes que producen lesiones tanto leves como graves. Este hecho se muestra en la gráfica de la Figura 1.

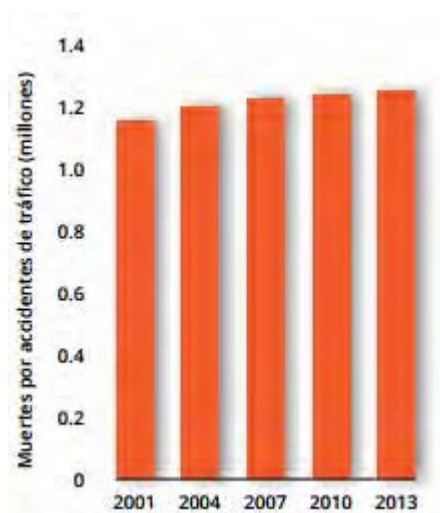


Figura 1: Muertes por accidente de tráfico en los últimos años (Fuente: www.who.int).

Estas alarmantes cifras de mortalidad en las carreteras hacen que las administraciones públicas estén actualmente muy interesadas en la seguridad vial. Además, los accidentes de conducción no solo repercuten en la salud de las personas, sino que también lo hacen a nivel económico, por lo que el hecho de evitar los accidentes de tráfico supone un beneficio doble.

Hoy en día, a fin de evitar estos efectos negativos, tanto en la vida de las personas como en la economía, son objeto de estudio y desarrollo los Sistemas Avanzados de Asistencia a la Conducción (*Advanced Driver Assistance Systems*, o *ADAS*). Los Sistemas Avanzados de Asistencia a la Conducción son sistemas desarrollados para automatizar, adaptar y mejorar los vehículos para una mayor seguridad y una mejor conducción [2].

El valor añadido de seguridad de estos sistemas reside en la posibilidad de advertir al conductor de potenciales situaciones de peligro mediante alarmas o tomando directamente el control del vehículo para evitar colisiones.

El grupo de investigación *CAOS* (Control, Aprendizaje y Optimización de Sistemas) de la Universidad Carlos III de Madrid está desarrollando un sistema basado en agentes centrado en un alto nivel de razonamiento que forma parte de un Sistema Avanzado de Asistencia a la Conducción [3]. Esta aproximación pone atención en la seguridad vial, en particular, en entornos urbanos.

Los agentes que se presentan dentro del *framework* tienen diferentes roles. Algunos de ellos tienen un rol relacionado con la recogida de información del tráfico (por ejemplo, la posición del peatón, el semáforo o las señales), del propio coche (por ejemplo, la velocidad o el ángulo de giro del volante), y del conductor (por ejemplo, hacia dónde mira). Otros agentes se dedican a la fusión y al análisis de los datos, que será lo que determinará si la situación actual de tráfico es peligrosa y decidirá si el sistema debe lanzar una alarma. Es en los agentes de este último grupo donde se implementará el sistema de toma de decisiones basado en reglas.

Este sistema multiagente trabaja en el entorno del simulador de conducción. El simulador utilizado es *STISIM Drive* [4], el cual permite simular, en tiempo real, la conducción normal de una persona, reproduciendo cualquier situación que se programe (ver Figura 2).



Figura 2: Plataforma de simulación basada en el simulador *STISIM Drive*.

Es importante el uso de un simulador, ya que para desarrollar el Sistema Avanzado de Asistencia a la Conducción es necesaria la realización de repetidos experimentos que representan situaciones peligrosas en la vida real, las cuales serían muy costosas y poco éticas de representar en la vida real. Se escoge este simulador de conducción frente a otros debido a que la mayoría de los simuladores de conducción tienen carencias. Algunas de las carencias se relacionan con que no representan fielmente el vehículo, no reproducen rutas reales y/o no dejan ampliar los escenarios que incluyen, o, lo más importante de todo, no dan la posibilidad de añadir inteligencia al simulador, es decir, poder programar en el propio simulador algoritmos que detecten objetos y/o proporcionen ciertos datos,

e incluso poder modificar la propia interfaz del simulador para poder mostrar avisos y demás información útil.

El simulador utilizado también permite obtener información generada durante el proceso de conducción, como por ejemplo, el ángulo de giro del volante en cada instante de tiempo, el nivel al que se están pisando los pedales de acelerador, embrague y freno, o a qué distancia se encuentra el vehículo de otro vehículo.

El grupo también dispone de una cámara *Kinect 2.0* para *Windows* (ver Figura 3). Este dispositivo aporta los datos relativos al conductor, tales como la posición de la cara y los ojos.



Figura 3: Cámara Kinect 2.0.

El problema que se plantea en este trabajo de fin de grado es la ampliación y mejora de la plataforma de simulación descrita. Por un lado se le dará a la plataforma de simulación la capacidad de simular escenarios reales mediante un módulo de gestión de rutas que los genere. Por otro lado, se incluirá en los agentes que se dedican a la fusión y al análisis de los datos que proporciona el simulador un sistema de toma de decisiones basado en reglas que detecte situaciones de peligro y reaccione ante ellas.

Como ya se ha mencionado, los accidentes de tráfico son un hecho muy triste y recurrente. En los últimos años ha aumentado el número de accidentes de tráfico en, aproximadamente, un millón de accidentes más [1].

Numerosos estudios han demostrado que aproximadamente el 78 % de los accidentes se deben a distracciones del propio conductor. La falta de atención del conductor puede ser causada por la fatiga o por la realización de actividades no relacionadas con la conducción, pero también puede ser debida a una situación de tráfico estresante donde múltiples situaciones de peligro pueden presentarse en un corto período de tiempo o espacio.

Un sistema incorporado en el vehículo que sea capaz de reconocer y evitar las situaciones de peligro o que el conductor esté distraído podría ser capaz de evitar a su vez numerosos daños e incluso salvar vidas. Por lo tanto, es interesante el desarrollo de *Sistemas Avanzados de Asistencia a la Conducción*.

El objetivo principal del trabajo de fin de grado es contribuir al desarrollo de un sistema multiagente que, a su vez, forma parte de un Sistema Avanzado de Asistencia a la Conducción. Más específicamente, el trabajo de fin de grado cubre dos objetivos principales.

El primero de los dos objetivos principales del proyecto es el desarrollo de un *sistema de toma de decisiones basado en reglas* que detecte situaciones peligrosas en ambientes urbanos y reaccione ante ellas disparando alarmas de aviso para el conductor. Las situaciones de peligro que se pueden detectar son, por ejemplo, un peatón cruzando por donde no debe o la excesiva cercanía al coche de delante. Este sistema formará parte de la aproximación mediante el sistema basado en agentes que se desarrolla en el grupo de investigación *CAOS*, que a su vez está incluida en un Sistema Avanzado de Asistencia a la Conducción.

El segundo de los dos objetivos principales del proyecto es el desarrollo de un *módulo de gestión de rutas* que se encargue de la traducción de coordenadas introducidas en el sistema de coordenadas *GPX* al lenguaje del simulador de conducción utilizado para el desarrollo del Sistema Avanzado de Asistencia a la Conducción *SDL* [43]. Con el desarrollo de este módulo, el simulador de conducción será capaz de realizar simulaciones en entornos reales, y no en circuitos o escenarios predefinidos o inventados.

Las coordenadas *GPX* se introducen mediante un fichero *XML* que contiene la ruta correspondiente en coordenadas geodésicas. Un ejemplo de fichero de ruta se presenta en la Figura 4.

El significado de cada una de las etiquetas se describe a continuación:

- **gpx**. Indica que es un fichero con coordenadas *GPX*. Adicionalmente, tiene un argumento, “creator”, para indicar el creador del archivo en cuestión.
- **metadata**. Señala el comienzo de la inclusión de metadatos del archivo.
- **copyright**. Describe el copyright del archivo, teniendo un argumento adicional, “author”, para indicar el autor del archivo en cuestión.
- **license**. Indica la licencia del archivo.
- **rte**. Indica el comienzo de la descripción de una ruta.
- **rtept**. Indica un punto de la ruta. Tiene dos argumentos, “lat” y “lon”, que sirven para indicar el valor de la latitud y la longitud de las coordenadas *GPX* del punto de ruta correspondiente.

```

<?xml version="1.0" encoding="UTF-8"?>
<gpx creator= - >
  <metadata>
    <copyright author= - >
      <license> - </license>
    </copyright>
  </metadata>
  <rte>
    <rtept lat="40.336291" lon="-3.770659"/>
    <rtept lat="40.336164" lon="-3.771785"/>
    <rtept lat="40.336108" lon="-3.772281"/>
    ...
  </rte>
</gpx>

```

Figura 4: Ejemplo de fichero de ruta *gpx*.

El lenguaje *SDL* del simulador de conducción consiste en la definición del escenario de simulación mediante la especificación de los diferentes eventos que tienen lugar según la distancia al punto de partida. Los ficheros que lee el simulador que están escritos en este lenguaje tienen extensión *.evt* o *.pde*. Cada uno de los posibles eventos se describe mediante una línea, indicando la distancia, el tipo de evento y los diferentes argumentos para cada uno de ellos. La definición de los diferentes eventos se puede encontrar en el manual de eventos del simulador [33].

Los objetivos específicos del trabajo de fin de grado, dividido en los dos sistemas descritos, son, por lo tanto:

1. Módulo de gestión de rutas.

- Extracción de información del recorrido a partir de ficheros de rutas expresadas en coordenadas *GPX*.
- Generación de un fichero de salida escrito en el lenguaje *SDL* del simulador con el escenario dado por la ruta.
- Diseño e implementación de una interfaz gráfica sencilla para configurar los parámetros de la aplicación de gestión de rutas final.

2. Sistema de toma de decisiones basado en reglas.

- Detección de situaciones peligrosas en ambientes urbanos.
- Reacción ante las situaciones peligrosas detectadas mediante alarmas para avisar al conductor.

Una vez especificados los requisitos del trabajo de fin de grado, diseñados e implementados los sistemas que cubren los objetivos descritos, se presentan los resultados obtenidos de ambos sistemas.

En primer lugar se presentan los resultados obtenidos del desarrollo del módulo de gestión de rutas. Este módulo se implementa como una aplicación *JAVA* con extensión *.jar*, la cual se puede ejecutar en la gran mayoría de sistemas operativos sin necesidad de ser compilada de nuevo.

Esta aplicación cuenta con una interfaz gráfica sencilla, que permite al usuario elegir la ruta del fichero de entrada, elegir la ruta del fichero de salida y permite iniciar el proceso de ejecución. En esta interfaz gráfica se tienen dos campos de texto con etiqueta y botón selector de ficheros asociados, y un tercer botón para iniciar la ejecución del sistema. En cada etiqueta se indica qué es lo que se debe introducir en cada campo de texto, y en el botón se puede navegar a través de los ficheros del computador en el que se ejecute y seleccionar directamente la ruta desde dicho navegador. Se muestra una captura de la interfaz gráfica en la Figura 5.

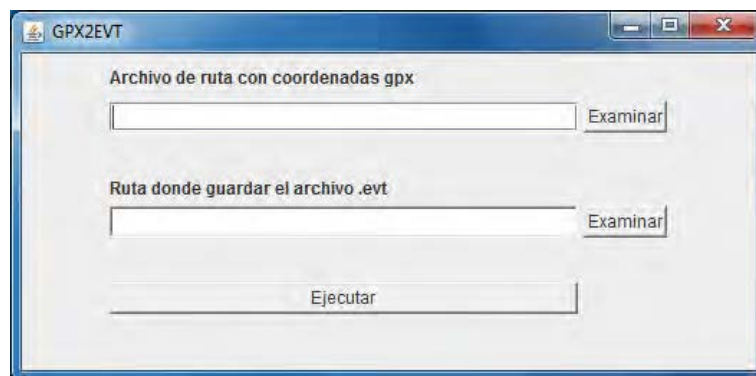


Figura 5: Interfaz del sistema - Gestión de rutas.

Una vez introducidos los parámetros de forma correcta, se ejecuta la aplicación, la cual procesa el fichero de coordenadas *GPX* introducido, extrayendo información relevante sobre la ruta que describe. El resultado de la ejecución es un fichero escrito en el lenguaje *SDL* del simulador con extensión *.evt*. Este fichero es capaz de ser leído y ejecutado por el simulador y contiene la ruta introducida como parámetros en el sistema de coordenadas *GPX*. De esta manera, el trazado que se recorre en el simulador corresponde con un recorrido que es el de una carretera real (ver Figura 6).



Figura 6: Fichero *SDL* ejecutado en el simulador de conducción.

Como la aplicación sólo procesa el recorrido de la ruta, no tiene ningún tipo de información acerca del entorno de la carretera. Esto hace que la apariencia de la vía sea la apariencia por defecto que ofrece el simulador.

En segundo lugar se presentan los resultados obtenidos del desarrollo del sistema de toma de decisiones basado en reglas. Este sistema es una ampliación del sistema multiagente ya implementado por el grupo de investigación CAOS de la Universidad Carlos III de Madrid [3], que forma parte del desarrollo de un Sistema Avanzado de Asistencia a la Conducción. Concretamente, este sistema está incluido en los agentes del sistema multiagente que se dedican a la fusión y el análisis de los datos, que es lo que determina si la situación del tráfico es peligrosa y decide, en caso afirmativo, si el sistema debe lanzar una alarma.

El sistema multiagente está implementado usando el *framework* de desarrollo de sistemas de agentes *JADE* (*JAVA Agent DEvelopment Framework*) [35]. Éste es un software enteramente implementado en el lenguaje de programación *JAVA* que simplifica el desarrollo de sistemas multiagentes. Por lo tanto, el desarrollo del sistema multiagente está codificado en lenguaje de programación *JAVA*, aunque se emplean otros lenguajes para la realización de tareas específicas. Estos lenguajes se integran mediante el uso de la librería *Apache Jena* [37].

Para el desarrollo del sistema de toma de decisiones se definen las situaciones y escenarios que se analizan. Para identificar estos escenarios, se realizó una encuesta a conductores, tanto noveles como con una experiencia de más de diez años con carnet de conducir, en la que se preguntaba acerca de situaciones de peligro en conducción por ciudad y acerca de la aceptación de un Sistema Avanzado de Asistencia a la Conducción que asista al conductor en esos casos. Tras este estudio y el análisis de los resultados de las encuestas, se definieron las siguientes cinco situaciones:

- **Situación 1 - Riesgo de colisión frontal.** El conductor se encuentra distraído en una carretera en la que existe tráfico, y la distancia al coche que precede al del conductor se recorta, llegando a ser muy pequeña.
- **Situación 2 - Riesgo de atropello.** Se está realizando una conducción por vía urbana y un peatón cruza la carretera a una distancia relativamente cercana al vehículo.
- **Situación 3 - Riesgo de colisión trasera.** Se va a realizar un adelantamiento, pero el coche de detrás del vehículo del conductor ya estaba adelantando.
- **Situación 4 - Riesgo de colisión lateral.** Se está realizando una conducción por vía urbana y un coche que estaba detenido sale sin señalizarlo.
- **Situación 5 - Riesgo de atropello de peatón sin visualizar.** Se está realizando una conducción por vía urbana y un peatón cruza la carretera saliendo de detrás de un vehículo aparcado o de un objeto que haga que el peatón no sea visible para el conductor.

Para el desarrollo del sistema de toma de decisiones también es necesario que se amplíe la ontología que se tenía [3] para incluir nuevos conceptos sobre la situación del tráfico y poder trabajar con ellos. Estos nuevos datos introducidos son:

- La existencia de vehículos en más posiciones (no sólo delante): detrás, detrás y a la izquierda, detrás y a la derecha, delante a la izquierda y delante a la derecha.
- El concepto del vehículo del conductor, con sus atributos asociados: freno, embrague, marcha, aceleración lateral, aceleración longitudinal, revoluciones por minuto, velocidad, ángulo de giro del volante y acelerador.

La ontología ampliada quedaría como se muestra en la Figura 7.

Una vez ampliada la ontología, se implementa la base de reglas del sistema de toma de decisiones. Para ello, se reúne la información necesaria de la ontología mediante consultas a la misma. Cuando ya se dispone de dicha información, se establecen las reglas necesarias para lanzar alarmas ante las situaciones definidas.

Como el sistema sólo tiene una salida, en el caso de que dos reglas se disparen para un mismo instante de tiempo, el sistema debería producir como salida aquella que indica un peligro mayor. Por lo tanto, se implementa también una jerarquía entre las diferentes alarmas disparadas por las reglas del sistema. Dicha jerarquía se expone en la tabla 1.

Tabla 1: Jerarquía de alarmas (ordenadas de mayor a menor prioridad).

Situación	Alarma	Prioridad
Situación 2	Riesgo de atropello	1
Situación 1	Riesgo de colisión frontal	2
Situación 5	Riesgo de atropello de peatón sin visualizar	3
Situación 4	Riesgo de colisión lateral	4
Situación 3	Riesgo de colisión trasera	5

Se considera el riesgo de atropello la situación más prioritaria, ya que está en juego la vida de una o más personas, y la vida es lo más prioritario. A continuación se considera el riesgo de colisión frontal seguido del atropello de un peatón sin visualizar. Se sigue este orden ya que no se puede alcanzar a los peatones no visualizados si hay un vehículo delante a poca distancia. Después se considera el riesgo de colisión lateral y, por último, el riesgo de colisión trasera, ya que solo se va a adelantar si no se produce ninguna otra situación de peligro.

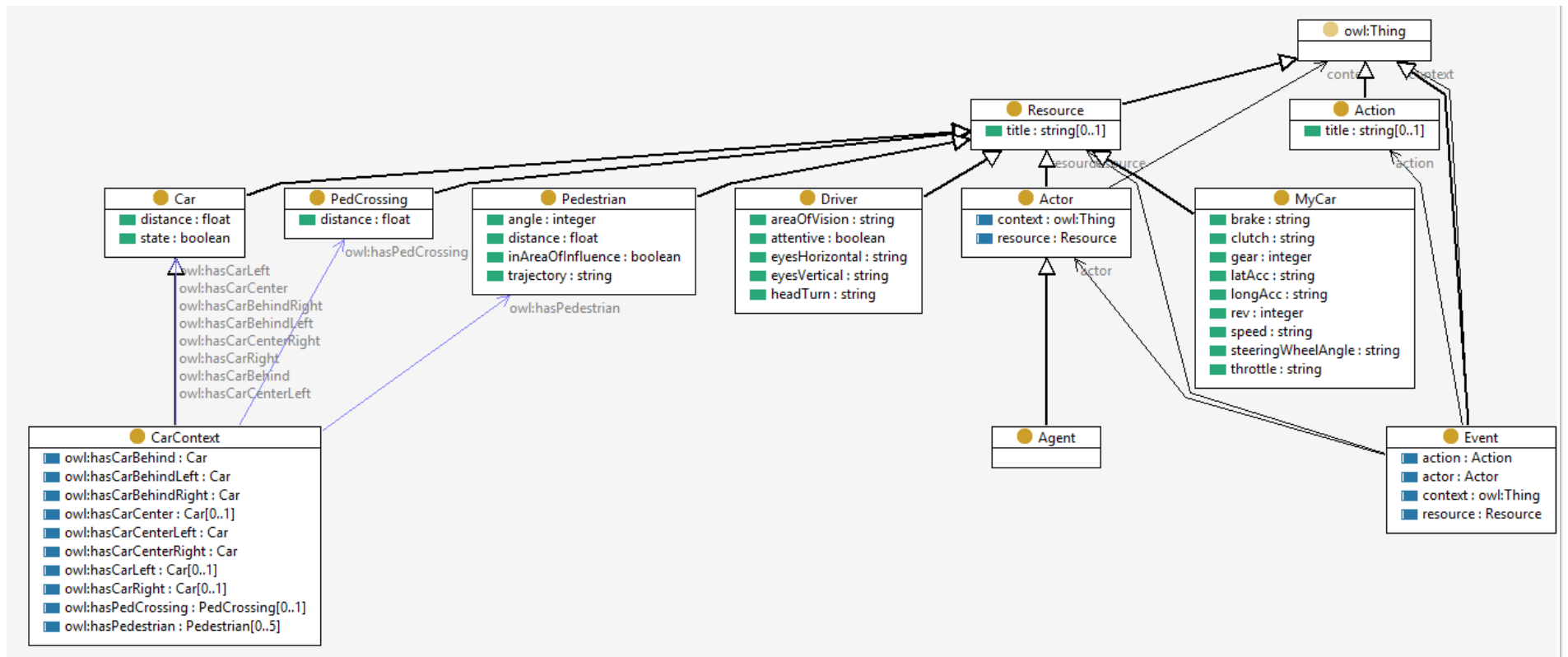


Figura 7: Ontología ampliada en el Trabajo de Fin de Grado.

En resumen, el resultado del sistema es la generación de una alarma que se lanza ante una situación de peligro dada mientras se conduce en el simulador de conducción. Se muestra un ejemplo de alarma lanzada ante una situación de peligro en la Figura 8.



Figura 8: Ejemplo de alarma “Riesgo de atropello” en el simulador de conducción.

En este caso, la alarma mostrada es la relativa a la situación de peligro 2: “Riesgo de atropello”.

Para finalizar el trabajo de fin de grado, se presentan las conclusiones. Se exponen las conclusiones extraídas de la realización del mismo, tanto técnicas como personales, y se presentan las líneas futuras de trabajo que se pueden seguir partiendo de este proyecto.

Tras la realización del presente proyecto, en cuanto a las conclusiones técnicas se concluye que los objetivos propuestos al comienzo del mismo han sido cumplidos. Por un lado se ha conseguido desarrollar el módulo de gestión de rutas capaz de leer e interpretar ficheros de rutas expresadas en coordenadas *GPX*, transformar dichas coordenadas al sistema de coordenadas *UTM* (*Universal Transversal de Mercator*), identificar el trazado a partir de esas coordenadas *UTM*, generar un fichero de salida escrito en lenguaje *SDL* del simulador con el escenario dado por la ruta, y que además cuenta con una interfaz gráfica que permite configurar los parámetros de la aplicación.

Por otro lado, se ha estudiado el sistema de agentes ya implementado, se ha ampliado la ontología existente para disponer de una información más completa por cada toma de muestra, y se ha conseguido desarrollar el sistema de toma de decisiones basado en reglas, que es capaz de realizar consultas sobre la ontología ampliada para reunir información y emplearla para implementar el sistema de reglas que lance alarmas.

De forma general, se destaca que el alumno ha sido capaz de aplicar sus conocimientos de una forma profesional a un problema relacionado con el ámbito de la ingeniería informática, reuniendo e interpretando datos relevantes dentro del área de estudio para emitir juicios que incluyen una reflexión sobre un tema relevante de índole científico, como es la seguridad vial y los sistemas avanzados de asistencia a la conducción.

Con respecto a las conclusiones personales extraídas, es satisfactorio y motivo de orgullo el haber realizado este proyecto que cierra mis estudios en el grado en Ingeniería Informática, puesto que se han cumplido todos los objetivos propuestos al principio.

Al realizar un proyecto software de tales dimensiones, se ha aprendido acerca de la dificultad y de la importancia de planificar el proyecto definiendo objetivos y tiempos de realización, e ir sacándolo adelante ajustándose a lo especificado. Además, la realización del proyecto ha supuesto un acercamiento al trabajo de investigación que es llevado a cabo por grupos de investigación como es el grupo con el que se ha trabajado, CAOS, y ha permitido al alumno desarrollar habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

También se realiza un aprendizaje en las propias tareas realizadas. Un ejemplo de esto es el aprendizaje del uso de la herramienta *TopBraid Composer ME* [45] para el manejo de la ontología, los lenguajes *RDF* [39], *OWL* [15], *SPARQL* [44] para la definición y consultas sobre esa ontología, o el lenguaje *SDL* [43] para la definición de escenarios del simulador de conducción utilizado.

Por último se exponen los trabajos futuros del trabajo de fin de grado. Este proyecto representa una primera aproximación a un sistema de toma de decisiones que forma parte de un sistema avanzado de asistencia a la conducción, por lo que el objetivo final del mismo es ir incorporado en un vehículo real. Como trabajo futuro principal destaca la posibilidad de completar el sistema aumentando el número de situaciones de peligro a considerar y advertir por el mismo, así como completar las situaciones ya existentes. Además, también se podrían diferenciar escenarios dentro de la misma situación, donde la gravedad aumenta a medida que la distancia disminuye.

El hecho de hacer que el sistema tenga en cuenta más situaciones lleva ligada la necesidad de ampliar la ontología para que ésta recoja más datos de la tarea de la conducción, tanto del conductor, como del vehículo y su entorno, y el sistema de reglas.

Una de las limitaciones del sistema es que los valores de los atributos de los conceptos son categóricos. Una mejora del sistema sería hacer que los datos que se obtienen del proceso de conducción sean numéricos, lo que haría que las reglas fueran mucho más precisas.

Una limitación más es que sólo se puede tener una salida, lo que implica la necesidad de establecer una jerarquía de alarmas, que corresponden a la salida del sistema. Si se amplía el número de situaciones a tener en cuenta, también se produce un aumento de la complejidad de dicha jerarquía. Es por eso que una de las líneas de trabajos futuros es la implementación del mismo sistema mediante el empleo de la lógica difusa, la cual permite considerar todas las reglas a la vez y prescindir así de la jerarquía de alarmas.

En cuanto al módulo de gestión de rutas implementado en este proyecto, éste tiene la limitación de que el sistema solo es capaz de leer los ficheros de rutas con un formato determinado. Como trabajo futuro se podría dar la capacidad al sistema de leer las rutas en más formatos y sistemas de coordenadas.

Otra limitación del sistema es que la salida del mismo sólo representa en el simulador la ruta a recorrer, pero no el entorno de la carretera. Es decir, se muestra el entorno de conducción que hay por defecto. Como trabajo futuro, se podría generar aleatoriamente un entorno con decoración alrededor y un estilo de vía acorde, tanto en su número de carriles y el sentido de los mismos como en su pavimentación.

Por último, también sería útil e interesante hacer que el sistema de gestión de rutas mostrara gráficamente, a partir de las coordenadas recibidas, el trazado de la ruta que se va a representar en lenguaje *SDL*.

Abstract

Road traffic safety is a worldwide very important issue. The share of population that dies each year by related causes has been increasing in the past few years. According to the *Global Status Report on Road Safety 2015* by World Health Organisation [1], road traffic injuries claim more than 1.25 million lives each year. That number doesn't include accidents that produces minor or severe injuries. This fact is shown in the chart of the Figure 9.

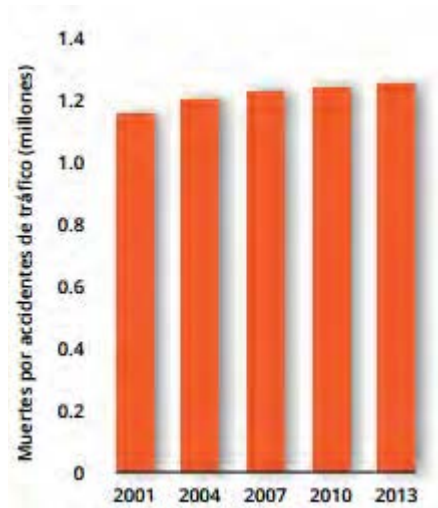


Figura 9: Death by traffic accident in recent years (Source: www.who.int).

These alarming numbers about road mortality make public administration very interested in road safety currently. Furthermore, road traffic accidents have an impact not only on people's health, but also on an economic level. Henceforth, the fact of avoiding road traffic accidents is translated into a twofold benefit.

Nowadays, in order to prevent these negatives impacts, both in people's lives and in economy, *Advanced Driver Assistance Systems (ADAS)* are subject of study and development. Advanced Driver Assistance Systems are systems developed to automatise, adapt and enhance vehicles for a better security and a better driving experience [2].

The added value of traffic security of these systems resides in the possibility of warning the driver about potential dangerous situations. This is done by using alarms or directly taking control of the vehicle in order to prevent collisions.

The investigation group *CAOS* (*Control, Aprendizaje y Optimización de Sistemas*, in English *Control Learning and Optimization Group*) at the University of Carlos III of Madrid is developing an agent-based system focused on high level reasoning as part of the development of Advanced Driver Assistance Systems [3]. This approach focuses on the driving safety, in particular, in urban environments.

The agents that are inside of the presented framework have different roles. Some of them have a role related to get information from the traffic (for example, pedestrian position, traffic lights or road signs), the car itself (for example, its speed or the steering wheel angle), and the driver (for example, where is he/she looking at). Other agents are involved in data fusion and its analysis. From this analysis, if a potentially risky traffic situation is not attended, the system will issue an alarm. The rules-based decision-making system is going to be implemented into this last group of agents.

This multi-agent system works within the driving simulator environment. The driving simulator used is *STISIM Drive* [4], which allows simulating, in real time, a person's usual driving, reproducing any situation that is programmed previously (see Figure 10).



Figura 10: Simulation platform based on STISIM Drive Simulator.

It is important to use a driving simulator, because it is needed to perform multiple experiments, that represents dangerous situations in real life, in order to develop the ADAS. These experiments would be very expensive and unethical if they were performed in real life. This driving simulator is chosen before the rest since most of the other driving simulators have shortcomings. Some of these shortcomings are that other simulators don't represent the vehicle accurately, don't reproduce real routes and/or don't allow expanding the included scenarios, or, most importantly of all, don't allow adding intelligence to the driving simulator. Adding intelligence means give the possibility of code in the driving simulator itself algorithms to track down objects and/or to obtain certain data, and even give the possibility of modify the own driving simulator interface in order to display warnings and other useful information.

The used driving simulator also allows obtaining data that is generated during the driving process. For instance, the steering wheel angle at each moment, the level of stepping on the throttle, clutch and break pedals, or the distance between this vehicle and the others.

The investigation group owns a *Kinect 2.0* camera for *Windows* (see Figure 11). This device provides data about the driver, such as the position of his/her face and the eyes.



Figura 11: Kinect 2.0 camera.

The problem that is raised on this end-of-degree project is the expansion and improvement of the described driving simulation platform. On one hand, the capacity to simulate the real scenarios by means of a route management module which generates them will be given to the driving simulation platform. On the other hand, a rules-based decision-making system which detects dangerous situations and reacts to them will be included on agents that are involved in data provided by the driving simulator fusion and its analysis.

As previously stated, traffic accidents are a sad, recurring fact. In the last few years the number of traffic accidents has been increased in, approximately, a million accidents more [1].

Many studies have shown that, approximately, 78% of the accidents are caused by distractions on the own driver. The lack of attention can be caused by drowsiness or activities not related to driving, although it can be due to a stressful traffic situation where multiple danger events can happen in a limited range of time or space.

A system embedded in the vehicle, which is able to detect and prevent dangerous situations or the driver distraction, could be able to prevent, at the same time, many damages and even save lives. Therefore, the development of an Advanced Driver Assistance System is interesting.

The main goal of the end-of-degree project is being a part of a multi-agent system that, at the same time, is a part of an Advanced Driver Assistance Systems. More particularly, the end-of-degree project covers two objectives.

The first of the two main objectives in this project is the development of a rules-based decision-making system which detects dangerous situations in urban environments and reacts to them by issuing alarms to warn the driver. The dangerous situations that can be prevented are, for instance, a pedestrian crossing improperly or excessive closeness to the vehicle in front. This system will be a part of the approach implemented with the agent based system that is developed by the investigation group *CAOS*, which, at the same time, is included in a Advanced Driver Assistance System.

The second of the two main objectives in this project is the development of a route management module that is responsible for the conversion of coordinates entered in the system as parameters in the GPX coordinate system to the language of the driving simulator used for the development of the Advanced Driver Assistance System *SDL*. With this module, the driving simulator would be able to simulate real routes instead of playing predefined or invented scenarios.

The *GPX* coordinates are introduced using a *XML* file which contains the corresponding route in geodesic coordinates. An example of the route file is represented in the Figure 12.

The meaning of each one of the labels is described below:

- **gpx**. It indicates that it is a file with *GPX* coordinates. Additionally, it has a “creator” argument to indicate the creator of the actual file.
- **metadata**. It indicates the beginning of the inclusion of metadata file.
- **copyright**. It describes the file copyright, having an additional argument, “author”, to indicate the actual file author.
- **license**. It indicates the file licence.
- **rte**. It indicates the beginning of the route description.
- **rtept**. It indicates a point in the route. It has two parameters, “lat” and “lon”, which are used to indicate the latitude and the longitude of the *GPX* coordinates of the corresponding route point.

```

<?xml version="1.0" encoding="UTF-8"?>
<gpx creator= - >
  <metadata>
    <copyright author= - >
      <license> - </license>
    </copyright>
  </metadata>
  <rte>
    <rtept lat="40.336291" lon="-3.770659"/>
    <rtept lat="40.336164" lon="-3.771785"/>
    <rtept lat="40.336108" lon="-3.772281"/>
    ...
  </rte>
</gpx>

```

Figura 12: Example of *GPX* route file.

The *SDL* language of the driving simulator consists in the definition of the simulation scenario through the specification of the different events which take place according to the distance to the starting point. The files which are read by the simulator and are written in this language have *.evt* or *.pde* extension. Each possible event is described in one line, indicating the distance to the starting point, the event type and the different parameters for each particular event. Instructions for defining different type of events can be found in the driving simulator event manual [33].

The specific objectives of the end-of-degree project, split in the two described systems, are, therefore:

1. Route management module.

- Path information extraction from route files expressed in *GPX* coordinates.
- Output file generation with the scenario given by the route, written in *SDL* language of the driving simulator.
- Simple graphic interface design and implementation to configure the parameters of the final route management application.

2. Rules-based decision-making system.

- Dangerous situations detection in urban environments.
- Dangerous situations reaction by issuing alarms in order to warn the driver.

Once the requirements for the end-of-degree project have been specified, and the systems that meet these objectives have been designed and implemented, it is shown the results obtained of both systems.

First of all, the results obtained about the route management module development are described. This module is implemented as a *JAVA* application with *.jar* extension. This allows the application to run in most of operative systems without being recompiled.

This application counts on a simple graphic interface, which allows the user to choose the path of the input file, to choose the path of the output file and to initiate the execution process. In this graphic interface there are two text fields with a label and a file selector button associated. In each associated label it is indicated what is the information that has to be entered in the corresponding text field. The associated button allows navigating through the computer files using a pop-up window and selecting the file path from this window directly. A screen capture of the graphic interface is shown in Figure 13.

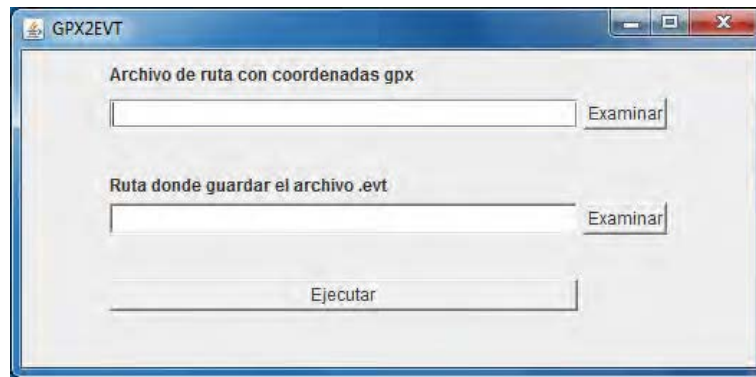


Figura 13: System graphic interface - Route management module.

Once the parameters have been entered, the application is run. It processes the entered *GPX* coordinates file, extracting relevant information about the route that it describes. The result of the execution is a file written in the driving simulator language, *SDL*, with *.evt* extension. This file can be read and executed by the driving simulator, and it contains the route entered as a parameter in *GPX* coordinates system. Thereby, the itinerary traversed on the driving simulator corresponds to a real road itinerary (see Figure 14).



Figura 14: *SDL* file executed on the driving simulator.

Since the application can only process the route path, it does not have any information about the road environment. This makes the road appearance be the default appearance that is offered by the driving simulator.

Secondly, the results obtained about the rules-based decision-making development are described. This system is an enlargement of the multiagent system that was already implemented by the investigation group CAOS at the University of Carlos III of Madrid [3], which is a part of the development of an Advanced Driver Assistance System. Specifically, this decision-making system is included on agents in the multi-agent system that are involved in data provided by the driving simulator fusion and its analysis. This analysis process determines if the traffic situation is dangerous and decides whether the system must issue an alarm or not.

The multi-agent system is implemented using the agent system development framework *JADE (JAVA Agent DEvelopment Framework)* [35]. This software is fully implemented in *JAVA* programming language, which simplifies the multi-agent system development. Therefore, the multi-agent development is coded in *JAVA* programming language, although other languages are used to specific tasks realisation. These languages are integrated using the *Apache Jena* library [37].

Situations and scenarios that are going to be analysed are defined to the decision-making system development. In order to identify these scenarios, drivers, both novice and experts, were surveyed. In this survey, there were questions about dangerous situations in an urban environment and about the acceptance of an Advanced Driver Assistance System that helps the driver in these cases. After the study and the analysis of the survey results, the following five situations were defined:

- **Situation 1 - Risk of frontal collision.** The driver is not attentive in a road where traffic is heavy, and the distance to the car in front is being shortened, becoming too short.
- **Situation 2 - Risk of running over.** While driving through an urban environment, a pedestrian crosses the road with a relatively short distance to the vehicle.
- **Situation 3 - Risk of rear collision.** An overtake is going to take place, but the car behind the driver's vehicle is currently overtaking.
- **Situation 4 - Risk of lateral collision.** While driving through an urban environment, a stopped car starts and begins to drive without any signalisation.
- **Situation 5 - Risk of running over a not visualised pedestrian.** While driving through an urban environment a pedestrian crosses the road from behind a stationed vehicle or an object that makes the pedestrian not visible to the driver.

For the development of the decision-making system is also necessary an expansion of the current ontology in order to include new and more complex concepts about the traffic situation, and with the purpose of being able to work with them. This new included data is:

- The existence of vehicles in more positions (not only in-front): behind, behind to the left, behind to the right, in-front to the left, in-front to the right.
- The concept of the driver's vehicle, with its associated attributes: brake pedal, clutch pedal, throttle pedal, gear, lateral acceleration, longitudinal acceleration, revolutions per minute, speed and steering wheel angle.

The new extended ontology is as shown on the Figure 15.

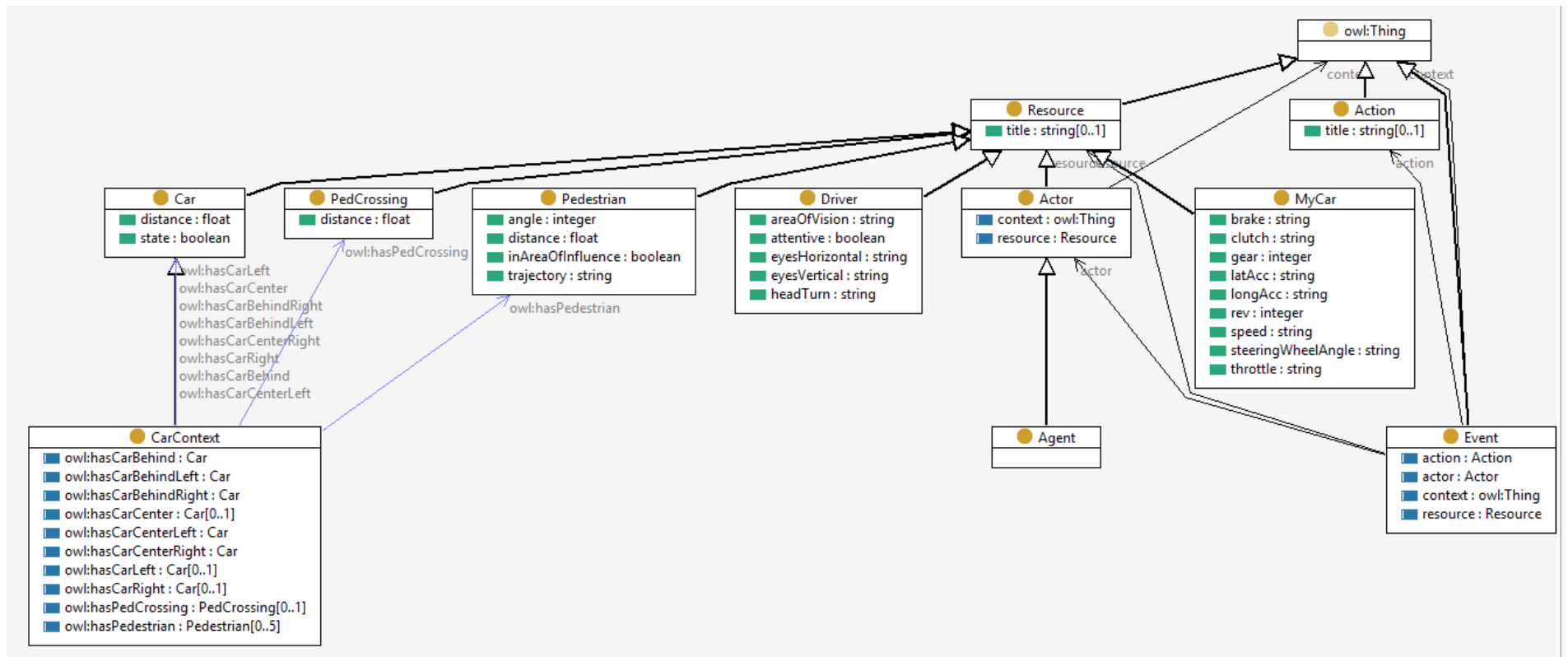


Figura 15: Extended ontology in the end-of-degree project.

Once extended the ontology, the base rules of the decision-making system are implemented. For this, the necessary data is gathered through ontology queries. Once this data is available, the necessary rules in order to issue alarms on defined situations are established.

Since the system has only one output, if two rules are activated at the same time, the system shall produce only the one with greater danger. Therefore, a hierarchy among the different issued alarms by the system rules is implemented. This hierarchy is exposed on Table 2.

Tabla 2: Alarm hierarchy (in order from highest to lowest priority).

Situation	Alarm	Priority
Situation 2	Risk of running over	1
Situation 1	Risk of frontal collision	2
Situation 5	Risk of running over a not visualised pedestrian	3
Situation 4	Risk of lateral collision	4
Situation 3	Risk of rear collision	5

The most prioritised situation is the risk of running over, as one or more lives are at stake, and life has the most priority. Next priority is the frontal collision, followed by running over a not visualised pedestrian. This order is followed as it is not possible to reach not visualised pedestrians if there is a close vehicle in-front. Then there is the lateral collision, and, finally, the risk of rear collision, as an overtake only takes place when there is apparently no other danger situation.

To sum up, the result of the system is an alarm generation that is issued when a danger situation happens while driving on the driving simulator. An example of an alarm issued on the presence of a dangerous situation is shown at the Figure 16.



Figura 16: Alarm “Risk of running over” example on the driving simulator.

In this case , the alarm is displayed on the dangerous situation 2: “Risk of running over”.

To finalise the end-of-degree project, the conclusions are presented, both technical and personal. Furthermore, future lines of work that can be done starting from this project are presented too.

Once the present project was completed, regarding the technical conclusions, it is concluded that the objectives presented at the start of the project have been accomplished. It was possible to develop the route management module which is able to read and interpret route files expressed in *GPX* coordinates, transform these coordinates to the *UTM* coordinate system, identify the outline from the *UTM* coordinates, generate an output file written on *SDL* language of the simulator with the scene given by the route, which also counts with a graphic interface that allows the configuration of the application parameters.

In addition, the previously implemented multi-agent system has been studied, the existent ontology has been extended in order to include a more complete information for each instance, and the rules-based decision-making system that is able to make queries about the ontology in order to gather data which will be used to implement the rules-based system that issue alarms has been developed.

Overall, it is distinguishable that the student has been able to apply his knowledge in a professional manner to a problem related to the field of computer science and engineering, gathering and interpreting relevant data within the study area to make judgements that include a reflection on a topic relevant to the scientific nature, such as road safety systems and Advanced Driver Assistance Systems.

Regarding the drawn personal conclusions, it is a matter of satisfaction to have carried out this project that closes my studies in the degree in Computer Science and Engineering, since it has met all the objectives set at the beginning.

When performing a software project of such dimensions, it has been learnt about the difficulty and the importance of planning the project, defining the objectives and completion times, and to take it forward within the specified plan. In addition, the project has meant an approach to a research work that is being carried out by a research group, *CAOS*, and which has allowed the student to develop the necessary learning skills to undertake further studies with a high degree of autonomy.

It is also learnt by doing the performed tasks. An example of this statement is the learning about using *TopBraid Composer ME* software tool [45], which is employed to manage the ontology. Another example is the learning about *RDF* [39], *OWL* [15], *SPARQL* for the ontology definition and queries, and *SDL* [43] language for the used driving simulator scenario definition.

To end, the future works of the end-of-degree project are exposed. This project represents a first approach to a decision-making system that is part of an advanced driving assistance, so that the ultimate objective is for it to be incorporated into a real vehicle. As a main future work, it is highlighted the possibility of completing the system by increasing the number of dangerous situations to take into consideration and to be warn by them, as well as to extend existing situations. Furthermore, different scenarios could also be differentiated on a same situation, where the gravity increases as the distance decreases.

The fact of making the system to take into account more situations leads the need to expand the ontology so that it collects more data from the driving task, both for the driver and the vehicle including its environment and the rules system.

One of the limitations of the system is that the values of the attributes of the concepts are categorical. An improvement of the system would be to make the data obtained from the driving process to be numeric, which would make the rules much more accurate.

A further limitation is that there can only have one output, implying the need of establishing a hierarchy of alarms, corresponding to the output of the system. If the number of situations to consider expands, this will increase the complexity of such hierarchy. That is why one of the future lines of work is the implementation of the same system using fuzzy logic, which allows to consider all the rules at once and to leave aside the hierarchy of alarms.

Regarding the route management module implemented in this project, it has the limitation that the system is only able to read the route files with a particular format. Future work could give the system the ability to read routes in more formats and coordinate systems.

Another limitation of the system is that the output represents only the route to go for the simulator, but not the road environment. That is, the default driving environment is shown. As future work, there is the possibility of randomly generating an environment with a style and decoration in line with it, both in number of lanes and their direction, as in the paving.

Finally, it would be useful and interesting to make the management system to graphically display the routes, from the received coordinates, plotting the route to be represented in *SDL* language.

Índice

1. Introducción	1
1.1. Motivación y objetivos	2
1.2. Descripción del problema	2
1.3. Alcance del proyecto	4
1.4. Restricciones	6
1.5. Marco regulador	7
1.6. Entorno operacional	7
1.7. Organización de la memoria	8
2. Estado del arte	10
2.1. Sistemas Avanzados de Asistencia a la Conducción	10
2.1.1. Sistemas Avanzados de Asistencia a la Conducción basados en el entorno	11
2.1.2. Sistemas Avanzados de Asistencia a la Conducción basados en el conductor	15
2.2. Simuladores de conducción	17
2.2.1. Simuladores de conducción empleados para la investigación	18
2.2.2. Simuladores de conducción empleados para la formación de conduc- tores	20
2.2.3. Simuladores de conducción empleados en el entretenimiento	23
3. Análisis del sistema	26
3.1. Especificación de requisitos - Gestión de rutas	26
3.1.1. Requisitos funcionales	27
3.1.2. Requisitos no funcionales	30
3.2. Casos de uso - Gestión de rutas	32
3.2.1. Descripción tabular de casos de uso	32
3.2.2. Descripción gráfica de casos de uso	34
3.3. Especificación de requisitos - Toma de decisiones	35
3.3.1. Requisitos funcionales	36
3.3.2. Requisitos no funcionales	39
3.4. Casos de uso - Toma de decisiones	40
3.4.1. Descripción tabular de casos de uso	40
3.4.2. Descripción gráfica de casos de uso	46
4. Arquitectura y diseño del sistema	48
4.1. Gestión de rutas	48
4.1.1. Módulo interfaz gráfica	49

4.1.2.	Módulo entrada-salida	49
4.1.3.	Módulo transformación de coordenadas	50
4.1.4.	Módulo tratamiento de coordenadas <i>UTM</i>	50
4.2.	Toma de decisiones	50
4.2.1.	Módulo gestión de consultas	52
4.2.2.	Módulo ontología	53
4.2.3.	Módulo razonamiento	53
5.	Implementación del sistema	54
5.1.	Gestión de rutas	54
5.1.1.	Entrada-Salida	54
5.1.2.	Transformación de coordenadas	55
5.1.3.	Tratamiento de coordenadas <i>UTM</i>	56
5.1.4.	Interfaz gráfica	59
5.2.	Toma de decisiones	59
5.2.1.	Ontología	63
5.2.2.	Gestión de consultas	68
5.2.3.	Razonamiento	69
6.	Resultados y evaluación	74
6.1.	Gestión de rutas	75
6.2.	Toma de decisiones	78
7.	Planificación del trabajo y entorno socioeconómico	85
7.1.	Metodología Software	85
7.2.	Planificación del trabajo	87
7.3.	Presupuesto	91
7.3.1.	Costes de personal	91
7.3.2.	Costes de material	92
7.3.3.	Costes totales	92
7.4.	Impacto socioeconómico esperado	93
7.5.	Impacto social y medioambiental	93
8.	Conclusiones y trabajos futuros	94
8.1.	Conclusiones técnicas	94
8.2.	Conclusiones personales y conocimientos adquiridos	95
8.3.	Trabajos futuros	95

Bibliografía

Anexos

- Anexo II: Manual de usuario - Sistema de Gestión de Rutas (uso de la aplicación)
- Anexo II: Encuestas acerca de situaciones de peligro en conducción por ciudad .

Índice de figuras

1.	Muertes por accidente de tráfico en los últimos años (Fuente: www.who.int).	I
2.	Plataforma de simulación basada en el simulador <i>STISIM Drive</i> .	II
3.	Cámara Kinect 2.0.	III
4.	Ejemplo de fichero de ruta <i>gpx</i> .	V
5.	Interfaz del sistema - Gestión de rutas.	VI
6.	Fichero <i>SDL</i> ejecutado en el simulador de conducción.	VI
7.	Ontología ampliada en el Trabajo de Fin de Grado.	IX
8.	Ejemplo de alarma “Riesgo de atropello” en el simulador de conducción.	X
9.	Death by traffic accident in recent years (Source: www.who.int).	I
10.	Simulation platform based on STISIM Drive Simulator.	II
11.	Kinect 2.0 camera.	III
12.	Example of <i>GPX</i> route file.	V
13.	System graphic interface - Route management module.	VI
14.	<i>SDL</i> file executed on the driving simulator.	VI
15.	Extended ontology in the end-of-degree project.	VIII
16.	Alarm “Risk of running over” example on the driving simulator.	IX
1.1.	Muertes por accidente de tráfico en los últimos años (Fuente: www.who.int).	1
1.2.	Plataforma de simulación basada en el simulador <i>STISIM Drive</i> .	3
1.3.	Cámara Kinect 2.0.	4
1.4.	Ejemplo de fichero de ruta <i>gpx</i> .	5
2.1.	Control de crucero adaptativo (Fuente: www.noticias.coches.com).	11
2.2.	Sistema de navegación GPS (Fuente: www.actualidadgadget.com).	12
2.3.	Control de luz adaptativo (ALC) (Fuente: www.circulaseguro.com).	12
2.4.	Reconocimiento automático de señales (Fuente: www.ford.es).	13
2.5.	Asistente de mantenimiento de carril (Fuente: www.circulaseguro.com).	13
2.6.	Frenado de emergencia autónomo (Fuente: www.euroncap.com).	14
2.7.	Sist. de asist. en aparcamiento (Fuente: www.aficionadosalamecanica.com).	14
2.8.	Asistente de vista lateral (Fuente: www.circulaseguro.com).	15
2.9.	Detección de expresiones faciales (Fuente: www.wired.com).	15
2.10.	Eyelid-Monitoring System (Fuente: www.cnet.com).	16
2.11.	Driver Workload Management (Fuente: blog.caranddriver.com).	16
2.12.	STISIM Drive Simulator (Fuente: www.stisimdrive.com).	18
2.13.	SIMESCAR (Fuente: www.simumak.com).	19
2.14.	NADS (Fuente: www.nads-sc.uiowa.edu).	20
2.15.	DriveSim (Fuente: www.drivesimsimulator.com).	21
2.16.	City Car Driving (Fuente: www.citycardriving.com).	22
2.17.	Sim. auto. Prado (Fuente: www.autoescuelaprado.es/simulador.html).	22

2.18. The Most Realistic Racing Simulator (Fuente: www.hammacher.com).	23
2.19. Forza Motorsport (Fuente: www.forzamotorsport.net).	24
2.20. Project Cars (Fuente: www.projectcarsgame.com).	25
3.1. Diagrama de casos de uso - gestión de rutas.	35
3.2. Diagrama de casos de uso - toma de decisiones.	47
4.1. Arquitectura del sistema - Gestión de rutas.	49
4.2. Arquitectura del sistema multiagente del simulador.	51
4.3. Arquitectura del sistema - Toma de decisiones.	52
5.1. Cálculo del radio de la curva.	56
5.2. Caso de ángulo 0°.	57
5.3. Caso de ángulo positivo.	58
5.4. Caso de ángulo negativo.	58
5.5. Interfaz del sistema - Gestión de rutas.	59
5.6. Situación 1 - Riesgo de colisión frontal (Fuente: saladeprensa.racc.es).	60
5.7. Situación 2 - Riesgo de atropello (Fuente: www.forotransportistas.es).	61
5.8. Situación 3 - Riesgo de colisión trasera (Fuente: www.circulaseguro.com).	61
5.9. Situación 4 - Riesgo de colisión lateral (Fuente: www.autofacil.es).	62
5.10. Situación 5 - Riesgo de atropello de peatón sin visualizar (Fuente: www.xataka.com).	62
5.11. Ontología iniciada por el grupo de investigación CAOS.	65
5.12. Ontología ampliada, sobre la del grupo de investigación CAOS, en este Trabajo de Fin de Grado.	67
5.13. Ejemplo de consulta SPARQL.	68
5.14. Imagen asociada a la alarma en la situación 1.	69
5.15. Imagen asociada a la alarma en la situación 2.	70
5.16. Imagen asociada a la alarma en la situación 3.	71
5.17. Imagen asociada a la alarma en la situación 4.	71
5.18. Imagen asociada a la alarma en la situación 5.	72
6.1. Prueba de riesgo de colisión frontal en el servicio web.	79
6.2. Prueba de peatón cruza en el simulador de conducción.	80
7.1. Planificación en cascada.	86
7.2. Planificación Gantt - Sistema de Gestión de Rutas.	89
7.3. Planificación Gantt - Sistema de Toma de Decisiones y Documentación.	90

Capítulo 1

Introducción

La seguridad en las carreteras es un tema que tiene gran importancia a nivel mundial. El porcentaje de población que muere cada año por causas relacionadas ha ido creciendo en los últimos años. Según el *Informe sobre la situación mundial de la seguridad vial 2015* de la *Organización Mundial de la Salud (OMS)* [1], 1,25 millones de personas mueren cada año por accidentes de tráfico, sin contar los accidentes que producen lesiones tanto leves como graves. Este hecho se muestra en la gráfica de la Figura 1.1.

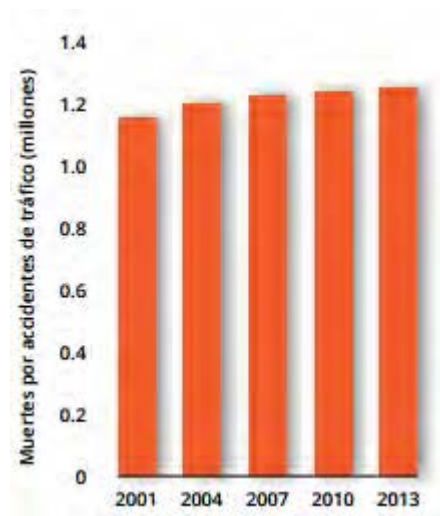


Figura 1.1: Muertes por accidente de tráfico en los últimos años (Fuente: www.who.int).

Estas alarmantes cifras de mortalidad en las carreteras hacen que las administraciones públicas estén actualmente muy interesadas en la seguridad vial. Además, los accidentes de conducción no solo repercuten en la salud de las personas, sino que también lo hacen a nivel económico, por lo que el hecho de evitar los accidentes de tráfico supone un beneficio doble.

Hoy en día, a fin de evitar estos efectos negativos, tanto en la vida de las personas como en la economía, son objeto de estudio y desarrollo los Sistemas Avanzados de Asistencia a la Conducción (*Advanced Driver Assistance Systems*, o *ADAS*). Los Sistemas Avanzados de Asistencia a la Conducción son sistemas desarrollados para automatizar, adaptar y mejorar los vehículos para una mayor seguridad y una mejor conducción [2].

El valor añadido de seguridad de estos sistemas reside en la posibilidad de advertir al conductor de potenciales situaciones de peligro mediante alarmas o tomando directamente el control del vehículo para evitar colisiones.

1.1. Motivación y objetivos

Como ya se ha mencionado, los accidentes de tráfico son un hecho muy triste y recurrente. En los últimos años ha aumentado el número de accidentes de tráfico en, aproximadamente, un millón de accidentes más [1].

Numerosos estudios han demostrado que aproximadamente el 78 % de los accidentes se deben a distracciones del propio conductor. La falta de atención del conductor puede ser causada por la fatiga o por la realización de actividades no relacionadas con la conducción, pero también puede ser debida a una situación de tráfico estresante donde múltiples situaciones de peligro pueden presentarse en un corto período de tiempo o espacio.

Un sistema incorporado en el vehículo que sea capaz de reconocer y evitar las situaciones de peligro o que el conductor esté distraído podría ser capaz de evitar a su vez numerosos daños e incluso salvar vidas. Por lo tanto, es interesante el desarrollo de *Sistemas Avanzados de Asistencia a la Conducción*.

El objetivo principal del trabajo de fin de grado es contribuir al desarrollo de un sistema multiagente que, a su vez, forma parte de un Sistema Avanzado de Asistencia a la Conducción.

1.2. Descripción del problema

El grupo de investigación *CAOS* (Control, Aprendizaje y Optimización de Sistemas) de la Universidad Carlos III de Madrid está desarrollando un sistema basado en agentes centrado en un alto nivel de razonamiento que forma parte de un Sistema Avanzado de Asistencia a la Conducción [3]. Esta aproximación pone atención en la seguridad vial, en particular, en entornos urbanos.

Los agentes que se presentan dentro del *framework* tienen diferentes roles. Algunos de ellos tienen un rol relacionado con la recogida de información del tráfico (por ejemplo, la posición del peatón, el semáforo o las señales), del propio coche (por ejemplo, la velocidad o el ángulo de giro del volante), y del conductor (por ejemplo, hacia dónde mira). Otros agentes se dedican a la fusión y al análisis de los datos, que será lo que determinará si la situación actual de tráfico es peligrosa y decidirá si el sistema debe lanzar una alarma. Es en los agentes de este último grupo donde se implementará el sistema de toma de decisiones basado en reglas.

Este sistema multiagente trabaja en el entorno del simulador de conducción. El simulador utilizado es *STISIM Drive* [4], el cual permite simular, en tiempo real, la conducción normal de una persona, reproduciendo cualquier situación que se programe (ver Figura 1.2).



Figura 1.2: Plataforma de simulación basada en el simulador *STISIM Drive*.

Es importante el uso de un simulador, ya que para desarrollar el Sistema Avanzado de Asistencia a la Conducción es necesaria la realización de repetidos experimentos que representan situaciones peligrosas en la vida real, las cuales serían muy costosas y poco éticas de representar en la vida real. Se escoge este simulador de conducción frente a otros debido a que la mayoría de los simuladores de conducción tienen carencias. Algunas de las carencias se relacionan con que no representan fielmente el vehículo, no reproducen rutas reales y/o no dejan ampliar los escenarios que incluyen, o, lo más importante de todo, no dan la posibilidad de añadir inteligencia al simulador, es decir, poder programar en el propio simulador algoritmos que detecten objetos y/o proporcionen ciertos datos, e incluso poder modificar la propia interfaz del simulador para poder mostrar avisos y demás información útil.

El simulador utilizado también permite obtener información generada durante el proceso de conducción, como por ejemplo, el ángulo de giro del volante en cada instante de tiempo, el nivel al que se están pisando los pedales de acelerador, embrague y freno, o a qué distancia se encuentra el vehículo de otro vehículo.

El sistema de simulación también dispone de una cámara *Kinect 2.0* para *Windows* (ver Figura 1.3). Este dispositivo aporta los datos relativos al conductor, tales como la posición de la cara y los ojos.



Figura 1.3: Cámara Kinect 2.0.

El problema que se plantea en este trabajo de fin de grado es la ampliación y mejora de la plataforma de simulación descrita. Por un lado se le dará a la plataforma de simulación la capacidad de simular escenarios reales mediante un módulo de gestión de rutas que los genere. Por otro lado, se incluirá en los agentes que se dedican a la fusión y al análisis de los datos que proporciona el simulador un sistema de toma de decisiones basado en reglas que detecte situaciones de peligro y reaccione ante ellas.

1.3. Alcance del proyecto

Como se ha descrito en la sección 1.2, este trabajo de fin de grado cubre dos objetivos principales.

El primero de los dos objetivos principales del proyecto es el desarrollo de un *sistema de toma de decisiones basado en reglas* que detecte situaciones de peligro en ambientes urbanos y reaccione ante ellas disparando alarmas de aviso para el conductor. Las situaciones de peligro que se pueden detectar son, por ejemplo, un peatón cruzando por donde no debe o la excesiva cercanía al coche de delante. Este sistema formará parte de la aproximación mediante el sistema basado en agentes que se desarrolla en el grupo de investigación *CAOS*, que a su vez está incluida en un Sistema Avanzado de Asistencia a la Conducción.

El segundo de los dos objetivos principales del proyecto es el desarrollo de un *módulo de gestión de rutas* que se encargue de la traducción de coordenadas introducidas en el sistema de coordenadas *GPX* al lenguaje del simulador de conducción utilizado para el desarrollo del Sistema Avanzado de Asistencia a la Conducción *SDL* [43]. Con el desarrollo de este módulo, el simulador de conducción será capaz de realizar simulaciones en entornos reales, y no en circuitos o escenarios predefinidos o inventados.

Las coordenadas *GPX* se introducen mediante un fichero *XML* que contiene la ruta correspondiente en coordenadas geodésicas. Un ejemplo de fichero de ruta se presenta en la Figura 1.4.

El significado de cada una de las etiquetas se describe a continuación:

- **gpx**. Indica que es un fichero con coordenadas *GPX*. Adicionalmente, tiene un argumento, “creator”, para indicar el creador del archivo en cuestión.
- **metadata**. Señala el comienzo de la inclusión de metadatos del archivo.
- **copyright**. Describe el copyright del archivo, teniendo un argumento adicional, “author”, para indicar el autor del archivo en cuestión.
- **license**. Indica la licencia del archivo.
- **rte**. Indica el comienzo de la descripción de una ruta.
- **rtept**. Indica un punto de la ruta. Tiene dos argumentos, “lat” y “lon”, que sirven para indicar el valor de la latitud y la longitud de las coordenadas *GPX* del punto de ruta correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>
<gpx creator= - >
  <metadata>
    <copyright author= - >
      <license> - </license>
    </copyright>
  </metadata>
  <rte>
    <rtept lat="40.336291" lon="-3.770659"/>
    <rtept lat="40.336164" lon="-3.771785"/>
    <rtept lat="40.336108" lon="-3.772281"/>
    ...
  </rte>
</gpx>
```

Figura 1.4: Ejemplo de fichero de ruta *gpx*.

El lenguaje *SDL* del simulador de conducción consiste en la definición del escenario de simulación mediante la especificación de los diferentes eventos que tienen lugar según la distancia al punto de partida. Los ficheros que lee el simulador que están escritos en este lenguaje tienen extensión *.evt* o *.pde*. Cada uno de los posibles eventos se describe mediante una línea, indicando la distancia, el tipo de evento y los diferentes argumentos para cada uno de ellos. La definición de los diferentes eventos se puede encontrar en el manual de eventos del simulador [33].

Los objetivos específicos del trabajo de fin de grado, dividido en los dos sistemas descritos, son, por lo tanto:

1. Módulo de gestión de rutas.

- Extracción de información del recorrido a partir de ficheros de rutas expresadas en coordenadas *GPX*.
- Generación de un fichero de salida escrito en el lenguaje *SDL* del simulador con el escenario dado por la ruta.
- Diseño e implementación de una interfaz gráfica sencilla para configurar los parámetros de la aplicación de gestión de rutas final.

2. Sistema de toma de decisiones basado en reglas.

- Detección de situaciones peligrosas en ambientes urbanos.
- Reacción ante las situaciones peligrosas detectadas mediante alarmas para avisar al conductor.

1.4. Restricciones

En esta sección de la memoria se especifican las restricciones, tanto hardware como software, para la realización del proyecto. Estas restricciones describen qué es lo necesario para el correcto desarrollo y ejecución de dicho proyecto.

Las restricciones de hardware del proyecto se muestran a continuación:

- Se deben disponer de tres monitores para poder simular de manera correcta el entorno de visión del conductor.
- La cámara que se usa para la detección facial del conductor debe ser una *Kinect 2.0* para poder utilizar el software adecuado.
- Se debe disponer de un juego de volante, pedales, palanca de cambios y asiento para realizar la simulación de la tarea de conducción de forma realista.
- Se deben de realizar las conexiones de forma adecuada entre todos los dispositivos, y se debe disponer de una estructura que los soporte.

Las restricciones de software del proyecto se muestran a continuación:

- El ordenador donde se conecta el simulador debe de contar con el software asociado (*STISIM Drive 3.03.08* o superior).
- Para poder ejecutar los programas escritos en *JAVA*, se debe disponer del *JDK* correspondiente.

1.5. Marco regulador

La finalidad del proyecto del que forma parte este trabajo de fin de grado es ir incorporado en un vehículo real y realizar la tarea de asistencia a la conducción, llegando incluso a tomar el control del vehículo. Como se trataría entonces de un vehículo de conducción autónoma y/o coexiste con otros vehículos de conducción autónoma, se debe conseguir la autorización de pruebas o ensayos de investigación realizados con vehículos de conducción automatizada en vías abiertas al tráfico en general, que se encuentra en la Instrucción 15/V-113 de la DGT [26] [27].

Además, dado que el sistema utiliza datos personales del usuario durante el funcionamiento del sistema (se monitoriza el rostro del conductor mediante una cámara de vídeo), será necesario cumplir con la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal [28].

1.6. Entorno operacional

En este apartado se describe el entorno operacional del proyecto, es decir, se citan las herramientas de software y hardware utilizadas durante el desarrollo del mismo.

El hardware empleado ha sido:

- Ordenador para ejecutar el simulador
 - Procesador Intel(R) Core(TM) i7-3770 CPU @ 3.4GHz
 - Memoria RAM de 16GB
 - SO Windows 7 Professional de 64 bits Service Pack 1
- Ordenador para el desarrollo del proyecto
 - Procesador Intel(R) Core(TM) i5-3470 CPU @ 3.2GHz
 - Memoria RAM de 8GB
 - SO Windows 7 Professional de 64 bits Service Pack 1
- Juego de volante de cuero, pedales y cambio de marchas Logitech G27

El software empleado ha sido:

- Simulador de conducción *STISIM Drive M100 Series* [4]
- Entorno de desarrollo *Java Eclipse Mars Release 4.5.1*
- Paquete ofimático de *Microsoft Office 2010 y 2013*
- Herramienta software *Microsoft Project 2010*
- Sistema de composición de textos *LaTeX*

- Herramientas web para la obtención de archivos de rutas con coordenadas *GPX*
 - *Open Source Routing Machine* [29]
 - *Open Route Service* [30]
 - *Google Maps to GPX converter* [31]
- *JAVA Agent DEvelopment Framework* (JADE)
- *TopBraid Composer Maestro Edition*
- Sistema de grabación y edición de vídeo *Camtasia*
- Herramienta software *Lucidchart*
- Herramienta software gratuita de dibujo matemático *GeoEnZo*.

1.7. Organización de la memoria

En esta sección se expone la estructura del documento, describiendo brevemente cada uno de los apartados de los que se compone.

En el Capítulo 1 se recoge la introducción al proyecto, que se compone de la motivación para su realización, la descripción del problema, el alcance del proyecto, las restricciones software y hardware, el marco regulador y el entorno operacional. Por último, se expone la organización del documento correspondiente a la memoria del mismo.

En el Capítulo 2 se realiza un análisis del estado del arte en los Sistemas Avanzados de Asistencia a la Conducción y en los simuladores de conducción.

En el Capítulo 3 se documenta el análisis de los sistemas que se van a implementar: el módulo de gestión de rutas y el sistema de toma de decisiones. Se especifican los requisitos y los casos de uso, tanto de forma tabular como gráfica, de cada uno de estos sistemas.

En el Capítulo 4 se expone la arquitectura y el diseño de los sistemas. Se realiza una descomposición en módulos y se describe la relación e interacción entre los módulos identificados dentro de cada uno de los dos sistemas, valorando durante el proceso diferentes soluciones alternativas.

En el Capítulo 5 se describe la implementación de los sistemas desarrollados en el trabajo de fin de grado, explicando el desarrollo de cada uno de los módulos especificados en el capítulo anterior.

En el Capítulo 6 se explican la experimentación y las pruebas realizadas en ambos sistemas para comprobar el correcto funcionamiento y para verificar que se cumplen los requisitos especificados en el capítulo 3.

En el Capítulo 7 se define la planificación del trabajo, la metodología de software seguida, el presupuesto para la realización del proyecto y un análisis del impacto socioeconómico y medioambiental esperado de la realización del proyecto. Se realiza un diagrama de *Gantt* para la planificación del proyecto, y un estudio de costes de personal y de materiales para los presupuestos del mismo.

En el Capítulo 8 se exponen las conclusiones técnicas y personales obtenidas con la realización del trabajo de fin de grado, así como las líneas futuras de trabajo y posibles mejoras que se pueden llevar a cabo a partir del mismo.

Para finalizar el documento, se incluye la bibliografía y referencias, así como los anexos.

Capítulo 2

Estado del arte

En este capítulo de la memoria se describe la tecnología más actual en el área de los *Sistemas Avanzados de Asistencia a la Conducción (ADAS)*, así como de los *simuladores de conducción* que se pueden encontrar en el mercado. Dentro de los Sistemas Avanzados de Asistencia a la Conducción, se diferenciará, como se hace en la bibliografía, entre los sistemas basados en el entorno y los basados en el conductor, y dentro de los simuladores de conducción, se diferenciará entre los empleados para la investigación, para la formación de conductores y para el entretenimiento.

2.1. Sistemas Avanzados de Asistencia a la Conducción

Los Sistemas Avanzados de Asistencia a la Conducción (*Advanced Driver Assistance Systems*), también conocidos como ADAS, son sistemas que se incluyen en los vehículos que asisten al conductor con el objetivo de evitar situaciones peligrosas y accidentes de tráfico. Éstos emplean información del entorno del vehículo que se recoge mediante sensores instalados en el vehículo.

Los ADAS aumentan la seguridad vial disminuyendo los errores en la conducción, a la par que aumentan la comodidad y mejoran la experiencia de conducción tanto para el conductor como para sus acompañantes. Es por eso que estos sistemas cada vez son más demandados, y eso ha hecho que los fondos dedicados a la investigación y el desarrollo de los mismos se hayan visto incrementados en los últimos años respecto de otros sistemas electrónicos o mecánicos de los vehículos. Existen dos tipos de ADAS, los basados en el entorno y los basados en el conductor.

2.1.1. Sistemas Avanzados de Asistencia a la Conducción basados en el entorno

El primer tipo, los **ADAS basados en el entorno**, son los que emplean información de la situación del vehículo y sus circunstancias para asistir la conducción. Son el tipo de ADAS más extendido desde el punto de vista comercial, apareciendo cada vez más, y no sólo en los vehículos de alta gama. Se exponen a continuación los principales ADAS basados en el entorno que se han incorporado a automóviles o se encuentran en fases muy avanzadas de desarrollo.

Control de crucero adaptativo (ACC)

Es un sistema que permite mantener una distancia de seguridad razonable con el vehículo delantero. Utiliza radares frontales para detectar el vehículo que está en el mismo carril y ajusta la velocidad del coche de forma automática [5] (ver Figura 2.1).

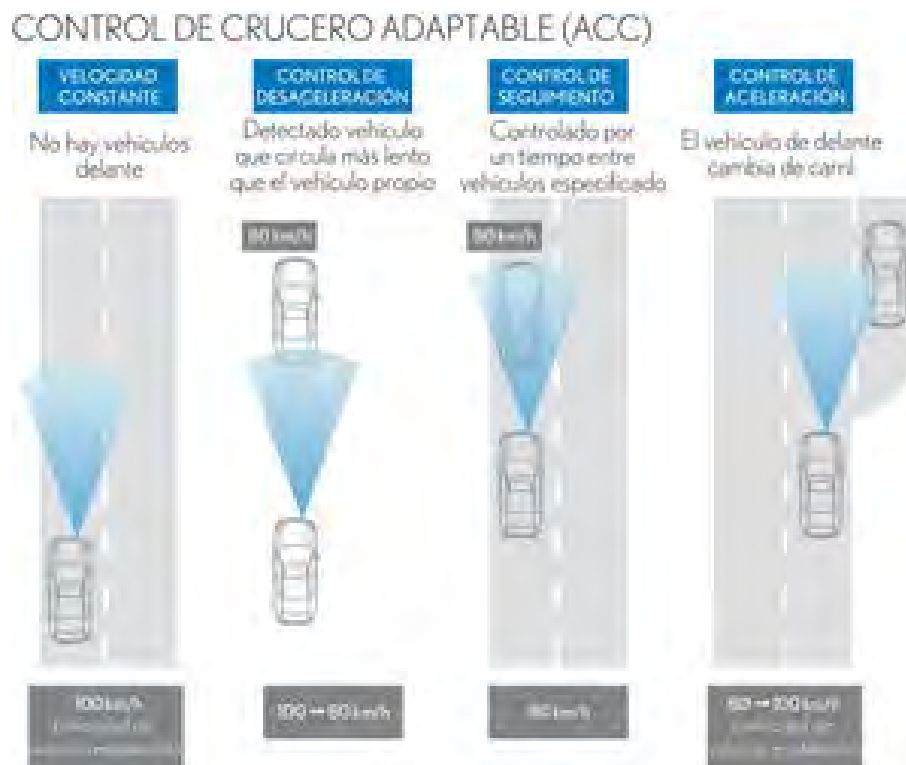


Figura 2.1: Control de crucero adaptativo (Fuente: www.noticias coches.com).

Sistema de navegación GPS

Es un sistema integrado en el vehículo que ubica y guía hacia su destino al conductor, tanto de forma gráfica como mediante comandos de voz, a través de una conexión vía satélite (ver Figura 2.2).



Figura 2.2: Sistema de navegación GPS (Fuente: www.actualidadgadget.com).

Control de luz adaptativo (ALC)

Es un sistema capaz de regular automáticamente la dirección de los faros del vehículo en las curvas para que el conductor disponga de una mejor visión de la carretera [6] (ver Figura 2.3).

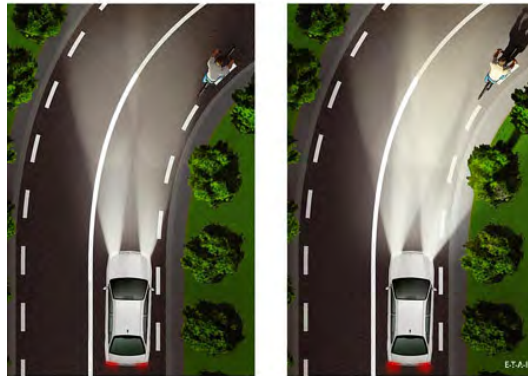


Figura 2.3: Control de luz adaptativo (ALC) (Fuente: www.circulaseguro.com).

Reconocimiento automático de señales

Consiste en una cámara situada, generalmente, en el interior del parabrisas que detecta de manera automática todas las señales, mostrándolas en el cuadro de mandos. Algunos de los **asistentes de velocidad límite permitida** se basan en este proceso, en el cual solo detectan las señales de control de velocidad [7] (ver Figura 2.4).



Figura 2.4: Reconocimiento automático de señales (Fuente: www.ford.es).

Asistente de mantenimiento de carril (LKA)

Es un sistema que advierte al conductor antes de que el vehículo abandone su carril de forma involuntaria (por ejemplo cuando el intermitente no está activo) mediante el empleo de una cámara ubicada en la parte superior del parabrisas [8] (ver Figura 2.5).

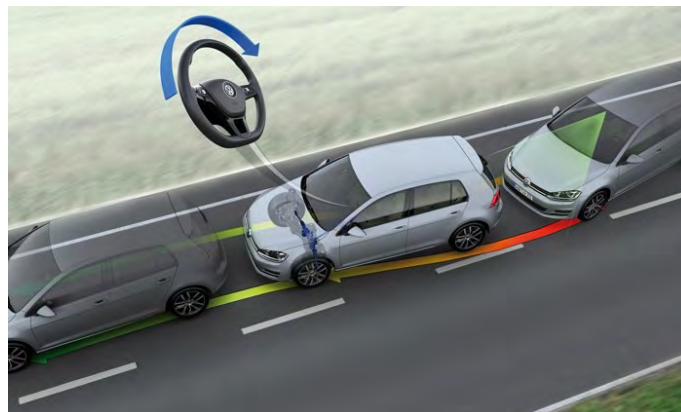


Figura 2.5: Asistente de mantenimiento de carril (Fuente: www.circulaseguro.com).

Frenado de emergencia autónomo (AEB)

Es un sistema que evita la situación de colisión frontal inminente. Para ello, el sistema detecta los objetos que están delante del vehículo empleando sensores frontales, y mide su velocidad relativa y su distancia respecto del mismo para determinar el frenado que se va a realizar [9] (ver Figura 2.6).



Figura 2.6: Frenado de emergencia autónomo (Fuente: www.euroncap.com).

Sistema de asistencia en aparcamiento

Mediante el uso de sensores de proximidad en los parachoques frontal y/o trasero, el sistema calcula la distancia a los objetos cercanos delante y/o detrás del vehículo para avisar al conductor con una señal sonora. Esta señal tiene una mayor frecuencia de repetición a medida que se va acercando el vehículo al objeto. En ocasiones, también se ofrece una visión del estado de los sensores en la pantalla central del vehículo [10] (ver Figura 2.7).



Figura 2.7: Sist. de asist. en aparcamiento (Fuente: www.aficionadosalamecanica.com).

Asistente de vista lateral

Es un sistema que permite detectar la presencia de objetos en el punto ciego del retrovisor del vehículo. Este sistema emplea sensores que detectan objetos a tres metros de distancia a los lados diagonalmente de la parte trasera del vehículo [11] (ver Figura 2.8).



Figura 2.8: Asistente de vista lateral (Fuente: www.circulaseguro.com).

2.1.2. Sistemas Avanzados de Asistencia a la Conducción basados en el conductor

El segundo tipo de ADAS, los **ADAS basados en el conductor**, son aquellos que integran la información del conductor para su tarea de asistencia a la conducción, con el fin de reducir los errores humanos. Estos ADAS son bien objeto de investigación, o puede encontrarse algún sistema en vehículos de muy alta gama. Se exponen a continuación los principales ADAS basados en el conductor:

Detección de expresiones faciales

El objetivo de este sistema es detectar estados físicos o anímicos del conductor, a través de su rostro, peligrosos para el desempeño de la tarea de conducción. Mediante sensores (por ejemplo una cámara), se trata de leer el estado facial para detectar el estado anímico del conductor, cuyas emociones perjudiciales para la conducción son cansancio, enfado, tristeza o distracción. Un ejemplo representativo es el sistema desarrollado por el fabricante de automóviles francés *Peugeot* en colaboración con *Transportation Center and Signal Processing 5 Laboratory of Ecole Polytechnique Fédérale de Lausanne (EPFL)* [12] (ver Figura 2.9).



Figura 2.9: Detección de expresiones faciales (Fuente: www.wired.com).

Eyelid-Monitoring System

El objetivo de este sistema es estudiar el estado de apertura de los ojos del conductor para emitir una alarma sonora si los párpados comienzan a cerrarse o si la frecuencia de parpadeo alcanza un nivel muy alto. Esto se realiza mediante una cámara que enfoca a la cara del conductor, la cual monitoriza los ojos. Un ejemplo representativo es el sistema desarrollado por el fabricante de coches japonés *Toyota* [13] (ver Figura 2.10).

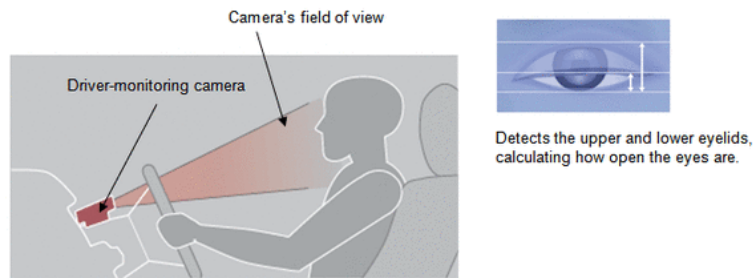


Figura 2.10: Eyelid-Monitoring System (Fuente: www.cnet.com).

Driver Workload Management

Se trata de un sistema inteligente de asistencia a la conducción que tiene como objetivo identificar patrones de cansancio o simples distracciones en el conductor que incrementan la posibilidad de sufrir un accidente de tráfico. Para ello, se emplean técnicas de Aprendizaje Automático sobre datos obtenidos de un simulador de conducción que permite la monitorización de las acciones del conductor. Ejemplos de los datos que se obtienen son el movimiento del volante, la presión de los pedales o la monitorización de los gestos faciales de la persona al volante. Este sistema se desarrolla en K. Torkkola et. al. [14] (ver Figura 2.11).



Figura 2.11: Driver Workload Management (Fuente: blog.caranddriver.com).

Sistema multiagente empleado en un ADAS en entornos urbanos

Este sistema multiagente se presenta en el artículo *Agent-Based Framework for Advanced Driver Assistance Systems in Urban Environments* escrito por Francisco Javier Ordoñez et. al. [3]. Se trata de una aproximación de un sistema basado en agentes con un alto nivel de razonamiento que forma parte de un ADAS. En este trabajo de fin de grado se ampliará este sistema como se explicó en las secciones 1.2 y 1.3.

Para representar los conceptos y sus relaciones acerca del entorno del tráfico urbano, se emplea una ontología *OWL* [15]. Esta ontología se ampliará en la realización de este trabajo de fin de grado, incluyendo nuevos conceptos e información acerca de conceptos ya existentes.

Esta aproximación tiene como objetivo la prevención de situaciones de peligro basándose en la fusión de la información del conductor, del vehículo y de su entorno, y está centrada en los entornos urbanos. Las situaciones de peligro se evitan mediante el aviso al conductor vía visual o sonora.

2.2. Simuladores de conducción

Un simulador es un sistema que reproduce el comportamiento de un sistema en ciertas condiciones, lo que permite que una persona pueda experimentar con él con el objetivo de entrenarse para situaciones que se dan en la vida real. De forma habitual, los simuladores combinan partes mecánicas o electrónicas y partes virtuales con el fin de generar una reproducción precisa de la realidad. Al tratarse de un simulador y al no poder sufrir daños reales la persona que lo utiliza, las situaciones que se pueden entrenar pueden ser de extremo peligro, con condiciones climáticas adversas, etc.

Los simuladores de conducción cada vez se emplean más y en más sitios para el entrenamiento de los conductores, tanto conductores de turismos, autobuses, como tractores o trenes. Este aumento de la utilización de simuladores ha llamado la atención de los investigadores de todo el mundo. Esto es debido a que, además de ser empleados para el entrenamiento de situaciones difíciles, peligrosas y poco comunes para los conductores, también permiten estudiar cómo reaccionan las personas y los vehículos ante las mismas. Estos usos permiten idear sistemas para evitar los daños que se pueden producir, reduciéndolos en gran medida, e incluso haciéndolos desaparecer. De esta manera, los simuladores de conducción permiten a los investigadores poner a los conductores en situaciones que serían ilegales y/o poco éticas, como, por ejemplo, la conducción con distracción del conductor, a fin de estudiar las reacciones, consecuencias y posibles soluciones ante éstas.

Además de su uso para entrenar a los conductores, tanto en conducción normal como en situaciones extremas, y para la investigación de ciertos eventos o escenarios de conducción, los simuladores también son empleados para el entretenimiento en videojuegos, donde los jugadores demandan cada vez más realismo. Además, debido a la fiel representación de los circuitos de carreras profesionales, estos simuladores también son empleados por pilotos para la memorización y familiarización de los trazados. A continuación se muestran los principales simuladores de conducción según el ámbito al que pertenecen.

2.2.1. Simuladores de conducción empleados para la investigación

En primer lugar, se exponen los simuladores de conducción empleados para la investigación. Como no existen simuladores de conducción diseñados exclusivamente para la investigación, los que se usan son empleados también con el objetivo formativo.

STISIM Drive Simulator

Se trata de un simulador de conducción de realidad virtual que es programable y totalmente interactivo. Tiene múltiples aplicaciones, tales como la investigación y el desarrollo, y la rehabilitación, evaluación y educación de conductores. Ofrece una serie de configuraciones a diferentes precios. La relación calidad-precio que ofrece es muy buena, y, según se va aumentando el precio de la configuración, se obtiene una experiencia de conducción más realista, por lo que se puede escoger la configuración que se adapte mejor a las necesidades de los investigadores [4] (ver Figura 2.12).

El sistema más avanzado que se puede obtener utilizando este simulador de conducción incluye, entre otras cosas:

- Gráficos de alta velocidad y procesamiento del sonido.
- Más de 80 escenarios prediseñados.
- El lenguaje de definición de escenarios (SDL) de STISIM Drive, que permite controlar absolutamente todo lo que ocurrirá en la simulación.
- Capacidad para modificar la interfaz mostrada en la simulación.
- Obtención de los datos de la simulación completa.
- Configuración del sistema de unidades utilizado en las medidas de velocidad, espacio, etc.
- Simulación de conducción en estado de embriaguez para concienciación en Seguridad Vial.



Figura 2.12: STISIM Drive Simulator (Fuente: www.stisimdrive.com).

SIMESCAR

Es un simulador de conducción de automóvil desarrollado por SIMUMAK [16] (ver Figura 2.13). Este simulador cumple con dos objetivos principales:

1. La formación de nuevos conductores mediante prácticas y repetición de ejercicios específicos.
2. La concienciación mediante prácticas de conducción a velocidades altas, distracciones al volante y conducción bajo los efectos del alcohol.

El sistema más avanzado que se puede obtener utilizando este simulador de conducción incluye, entre otras cosas:

- Simulación de conducción en estado de embriaguez para concienciación en Seguridad Vial.
- Variedad de escenarios: montaña, autopista, ciudad, etc. no modificables.
- Variedad de puntos de inicio y de entornos específicos dentro de cada escenario seleccionado.
- Conducción con condiciones meteorológicas distintas y de intensidad modificable.
- Parametrización de intensidad de agresividad del resto de coches y/o peatones del entorno.



Figura 2.13: SIMESCAR (Fuente: www.simumak.com).

The National Advanced Driving Simulator (NADS)

Se trata del simulador de conducción más sofisticado del mundo, y está desarrollado por la *NHTSA* (*National Highway Traffic Safety Administration*). El NADS ofrece una simulación de conducción de alta fidelidad a tiempo real. Consiste en una plataforma con una gran cápsula en cuyo interior se construye un vehículo, que puede ser un turismo, un camión o un autobús. La cabina del vehículo está equipada electrónicamente y mecánicamente con instrumentación específica que hace que la cabina se desplace, rote, gire, vibre, etc. (teniendo 13 grados de libertad en el movimiento), haciendo que la simulación sea mucho más realista. Esto se une al hecho de que el interior de la cápsula reproduce el entorno de la conducción en una pantalla curva de 360° alrededor del vehículo, disponiendo además de un equipo de sonido para dar el máximo realismo posible [17] (ver Figura 2.14).

En este entorno, el conductor se somete a diferentes situaciones de conducción, sin ser expuesto a ningún tipo de peligro, produciendo datos que serán analizados posteriormente por un equipo de investigación. Para este propósito, la NHTSA también dispone de alternativas un poco más simples como el NADS-2 y el miniSim.



Figura 2.14: NADS (Fuente: www.nads-sc.uiowa.edu).

2.2.2. Simuladores de conducción empleados para la formación de conductores

A continuación, se presentan los simuladores de conducción utilizados exclusivamente para la formación de conductores:

DriveSim

Se trata de un programa de simulación de conducción desarrollado por la empresa *DriveSim Simulation S.A.* que permite practicar la circulación como si se tratara de conducir un vehículo real (ver Figura 2.15). El simulador de conducción DriveSim ha sido especialmente diseñado para autoescuelas, centros de formación, seguridad vial y eventos, como una herramienta educativa [18].

Este simulador de autoescuela representa todas aquellas situaciones que se pueden encontrar circulando por ciudad, vías interurbanas, autopistas o caminos, pudiendo personalizar la conducción según las necesidades educativas. Por ejemplo, se puede conducir a diferentes horas del día, con diferente climatología, con o sin ABS, en superficies deslizantes, etc.



Figura 2.15: DriveSim (Fuente: www.drivesimsimulator.com).

City Car Driving

Este simulador de conducción, conocido anteriormente como 3D Instructor 2.0, es un simulador de autoescuela que, junto con el hardware adecuado (tiene soporte para realidad virtual mediante *Oculus Rift* [19]), permite al futuro conductor conducir por una gran ciudad de forma guiada (ver Figura 2.16). La hora del día y la meteorología en la conducción va cambiando a medida que la simulación avanza. Se tienen como posibles momentos del día la tarde, la mañana, anocheciendo y la noche, y como posibles valores de meteorología tiempo seco, despejado, nublado, húmedo, lluvioso y niebla [20].

En el escenario simulado se pueden encontrar tranvías, peatones y tráfico controlado por el propio simulador de manera inteligente, pudiendo los vehículos chocar unos con otros o contra el usuario. Además, se dispone de una serie de misiones de conducción con diferentes niveles de dificultad que tienen el objetivo de entrenar al futuro conductor que utiliza el simulador de conducción.



Figura 2.16: City Car Driving (Fuente: www.citycardriving.com).

Simulador propio

Algunas autoescuelas disponen de su propio simulador donde los futuros conductores pueden entrenarse antes de conducir un coche por primera vez. El empleo del simulador no pretende eliminar las prácticas con el coche, sino que sirve para iniciarse en el manejo de los mandos del aparato sin miedo a cometer errores ni al tráfico para un mejor aprovechamiento de las clases [21] (ver Figura 2.17).

Los alumnos pueden practicar en el simulador la conducción por carretera, vías urbanas, adelantamientos, estacionamiento, marcha atrás y numerosas situaciones más. El simulador dispone de dos modos: el modo enseñanza y el modo libre. El modo enseñanza tiene cinco bloques didácticos con ejercicios en los que se analiza una situación concreta de la conducción, mientras que en el modo libre se elige el coche, el tipo de vía y las condiciones atmosféricas y se conduce libremente, pudiendo practicar la conducción eficiente, adelantamientos en vías de doble sentido, incorporaciones, etc.



Figura 2.17: Sim. auto. Prado (Fuente: www.autoescuelaprado.es/simulador.html).

2.2.3. Simuladores de conducción empleados en el entretenimiento

Por último, se describen los simuladores de conducción para el entretenimiento. Éstos simulan circuitos de carreras empleados, además de para el entretenimiento, para la utilización de pilotos profesionales para practicar los trazados de los circuitos:

The Most Realistic Racing Simulator

Distribuido únicamente por la compañía estadounidense de tiendas de retail y venta por catálogo *Hammacher Schlemmer*, este simulador de carreras es el más realista y más caro (185.000\$ actualmente), de todos los que hay en el mercado. Ha sido seleccionado por la compañía fabricante de coches estadounidense *Ford* para la demostración de las experiencias de viaje [22].

Dispone de servo actuadores lineales para la suspensión del coche, chasis monocasco de fibra de vidrio para girar, inclinar y rotar el dispositivo 360 grados hasta una aceleración de 0,5G. Además, como medidas de seguridad, cuenta con un asiento de carreras real, cinturón de seguridad y “puertas”. Por último, para ofrecer la máxima experiencia de realismo y para ofrecer una conducción ágil como en las carreras de la vida real, dispone de un volante donde se pueden cambiar las marchas y pedales de acelerador, freno y embrague.

El sistema incluye la simulación de doce coches donde se encuentran modelos GT, F1 y F3, entre otros, y 16 circuitos, desde *Joesville* hasta *Nuerburg*, que se muestran en su monitor HD triple con un sistema de audio de 500W (ver Figura 2.18).



Figura 2.18: The Most Realistic Racing Simulator (Fuente: www.hammacher.com).

Forza Motorsport

Se trata de una saga de videojuegos de carreras y simulación automovilística, que ya posee 6 títulos, desarrollada por *Turn 10 Studios* y publicada por *Microsoft Studios*. En el último de sus títulos hasta la fecha, lanzado a finales de 2015, se incluyen 460 vehículos y 26 circuitos en total [23].

Ya sea con el mando de la consola o con un sistema de conducción instalado (volante, palanca de cambios y pedales de aceleración, frenado y embrague), se percibe que cada coche que se puede pilotar ha sido perfectamente recreado a nivel técnico (ver Figura 2.19). Sin embargo, el volante ofrece una mayor precisión en todas las sensaciones que transmite el coche, que ofrece un diferente comportamiento, vibración y resistencia en las diferentes superficies de cada trazado (hay 148 superficies diferentes en el simulador).

También se simulan las averías que puede sufrir el coche, tanto visuales, en el aspecto del vehículo, como de rendimiento del mismo. Estas averías afectan notablemente a la conducción del vehículo. Además, se simulan los efectos en la dinámica del coche a causa de efectos meteorológicos como el frío o la lluvia, incluso simulando el *aquaplaning* en los charcos y otros muchos detalles más.



Figura 2.19: Forza Motorsport (Fuente: www.forzamotorsport.net).

Project CARS (Community Assisted Racing Simulator)

Es un videojuego simulador de carreras desarrollado por *Slightly Mad Studios* y la comunidad online de *World of Mass Development* y lanzado el 7 de mayo de 2015. Este simulador de carreras cuenta con un centenar de vehículos, en su mayoría reales, pertenecientes a marcas como *Audi*, *BMW*, *Caterham*, *Lotus*, *McLaren*, *Mercedes-Benz*, *Pagani* o *RUF*, y sesenta circuitos reales en su inmensa mayoría: *Mónaco*, *Montmeló*, *Brands Hatch*, *Hockenheim*, etc. [24].

En su desarrollo se ha puesto mucho énfasis en el realismo, de modo que han simulado detalles como los daños mecánicos o la temperatura de los neumáticos. Lo que más destaca es la implementación del ciclo día-noche, que hace que la iluminación pueda variar a lo largo de las carreras, llegando a conducir de noche; y la meteorología dinámica, que hace que, por ejemplo, la lluvia aparezca y desaparezca. Además, el simulador es compatible con dispositivos de realidad virtual como *Playstation VR* (anteriormente conocido como *Project Morpheus*) [25] y *Oculus Rift* [19], que harán sentir al conductor como si estuviera en el interior de la cabina de un vehículo.



Figura 2.20: Project Cars (Fuente: www.projectcarsgame.com).

Capítulo 3

Análisis del sistema

Como ya se mencionó en la sección 1.3, este proyecto cubre dos objetivos, por lo que se compone de dos sistemas: un módulo de gestión de rutas para el simulador y un sistema de toma de decisiones basado en reglas. Por consiguiente, en este capítulo de la memoria se realizan dos análisis, con el fin de obtener una especificación de requisitos detallada de los sistemas que satisfaga las necesidades de información de los usuarios y que sirva de base para el posterior diseño del sistema. En primer lugar, se especificarán los requisitos funcionales y no funcionales que debe cumplir el sistema de gestión de rutas y los distintos casos de uso de dicho sistema en su representación tabular y gráfica. A continuación, se procede de igual forma con el sistema de toma de decisiones.

3.1. Especificación de requisitos - Gestión de rutas

Los requisitos para esta parte del proyecto son obtenidos directamente desde el grupo de investigación, el cual quiere generar rutas para su simulador a partir de archivos de coordenadas *GPX*. De esta manera, las rutas generadas son trazados reales generados automáticamente.

Para realizar la especificación de los requisitos del sistema de gestión de rutas se emplea una tabla para cada uno de ellos con el siguiente formato (tabla 3.1):

Tabla 3.1: Modelo para requisitos.

Identificador	RX - YY		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción			

A continuación se explica qué significa cada una de las filas de la tabla:

- **Identificador.** Representa una manera de reconocer unívocamente al requisito en cuestión para facilitar la trazabilidad. Los caracteres del identificador poseen el siguiente significado:
 - **R.** Requisito.
 - **Y.** Puede tomar los valores F para requisitos funcionales o NF para requisitos no funcionales.
 - **X.** Número entero único para cada requisito funcional o no funcional.
- **Prioridad.** Indica la prioridad de cumplimiento del requisito en tres niveles: prioridad alta, prioridad media o prioridad baja. Es útil para decidir la planificación del proyecto.
- **Necesidad.** Especifica la importancia del requisito para el desarrollo del proyecto en tres niveles: necesidad esencial, necesidad deseable o necesidad opcional. Que un requisito tenga necesidad esencial quiere decir que su cumplimiento es obligatorio, y en caso de no cumplirse el proyecto no podría ser llevado a cabo. Que un requisito sea deseable quiere decir que su cumplimiento sería bueno para el proyecto, pero no se asegura su cumplimiento por algún motivo. Y, por último, que un requisito tenga necesidad opcional quiere decir que puede o no ser cumplido sin afectar en gran medida al proyecto en su totalidad.
- **Estabilidad.** Representa si el requisito puede cambiar, o no cambiar, durante el desarrollo del proyecto en tres niveles: estabilidad alta, estabilidad media o estabilidad baja.
- **Descripción.** En esta fila de la tabla se realiza una descripción breve pero detallada del requisito en cuestión.

3.1.1. Requisitos funcionales

Los requisitos funcionales especifican qué tiene que hacer el software, describiendo cada uno de ellos una funcionalidad del sistema a desarrollar. Los requisitos funcionales de esta primera parte del proyecto se presentan a continuación (de la Tabla 3.2 a la 3.14):

Tabla 3.2: RF - 01.

Identificador	RF - 01		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema leerá ficheros de ruta de coordenadas <i>GPX</i> con extensión <i>.gpx</i> .		

Tabla 3.3: RF - 02.

Identificador	RF - 02		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema transformará las coordenadas <i>GPX</i> en coordenadas <i>UTM</i> (<i>Universal Transversal de Mercator</i>).		

Tabla 3.4: RF - 03.

Identificador	RF - 03		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema calculará la distancia en metros entre cada par de puntos.		

Tabla 3.5: RF - 04.

Identificador	RF - 04		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema será capaz de calcular el radio de las curvas.		

Tabla 3.6: RF - 05.

Identificador	RF - 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema será capaz de calcular la distancia recorrida en el arco de las curvas.		

Tabla 3.7: RF - 06.

Identificador	RF - 06		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema será capaz de calcular el sentido de giro de las curvas.		

Tabla 3.8: RF - 07.

Identificador	RF - 07		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema contará con un método que calcula el ángulo que forman dos vectores de dos dimensiones.		

Tabla 3.9: RF - 08.

Identificador	RF - 08		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema identificará el trazado de la ruta a partir de las coordenadas recibidas.		

Tabla 3.10: RF - 09.

Identificador	RF - 09		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema escribirá en un fichero con extensión .evt la ruta en lenguaje SDL.		

Tabla 3.11: RF - 10.

Identificador	RF - 10		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema contará con una interfaz gráfica para su uso.		

Tabla 3.12: RF - 11.

Identificador	RF - 11		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz gráfica del sistema permitirá seleccionar la ruta del fichero de coordenadas <i>GPX</i> .		

Tabla 3.13: RF - 12.

Identificador	RF - 12		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz gráfica del sistema permitirá seleccionar la ruta de destino donde será almacenado el fichero con la ruta en lenguaje SDL.		

Tabla 3.14: RF - 13.

Identificador	RF - 13		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz gráfica del sistema mostrará un mensaje en la pantalla cuando el proceso haya finalizado.		

3.1.2. Requisitos no funcionales

Los requisitos no funcionales de un sistema, en vez de describir qué debe hacer el sistema, describen características de funcionamiento del mismo. Los requisitos no funcionales de esta primera parte del proyecto se presentan a continuación (de la tabla 3.15 a la 3.20):

Tabla 3.15: RNF - 01.

Identificador	RNF - 01		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema trabajará en el sistema métrico.		

Tabla 3.16: RNF - 02.

Identificador	RNF - 02		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Descripción	El sistema será implementado en el lenguaje de programación JAVA.		

Tabla 3.17: RNF - 03.

Identificador	RNF - 03		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Descripción	El sistema se distribuirá finalmente como una aplicación JAVA con extensión .jar		

Tabla 3.18: RNF - 04.

Identificador	RNF - 04		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El usuario final debe disponer de la ruta en un fichero de coordenadas con extensión .gpx con las coordenadas GPX en formato XML.		

Tabla 3.19: RNF - 05.

Identificador	RNF - 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El usuario deberá seleccionar en la interfaz gráfica la ruta del fichero de coordenadas GPX con extensión .gpx.		

Tabla 3.20: RNF - 06.

Identificador	RNF - 06		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El usuario deberá seleccionar en la interfaz gráfica la ruta de destino para almacenar el fichero con extensión .evt con la ruta en lenguaje SDL.		

3.2. Casos de uso - Gestión de rutas

Ahora se procede a definir los casos de uso de la primera parte del proyecto: el módulo de gestión de rutas para el simulador. Los casos de uso son la descripción de los pasos que debe seguir un actor, es decir, el personaje que realiza la acción, para la realización de una tarea concreta en el sistema. Se pueden describir de manera tabular, es decir, mediante una tabla, o de manera gráfica, es decir, mediante un diagrama (diagrama de caso de uso).

3.2.1. Descripción tabular de casos de uso

Para especificar los casos de uso en su forma tabular, se usan tablas siguiendo el siguiente formato:

Tabla 3.21: Modelo para casos de uso.

Identificador	CU - XX
Título	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario de éxito	

A continuación se explica qué significa cada una de las filas de la tabla:

- **Identificador.** Representa una manera de reconocer unívocamente al caso de uso en cuestión para facilitar la trazabilidad. Los caracteres del identificador poseen el siguiente significado:
 - **CU.** Caso de uso.
 - **XX.** Número entero único para cada caso de uso.
- **Título.** Nombre con el que se identifica al caso de uso.
- **Objetivo.** Lo que se pretende conseguir con el caso de uso que se está definiendo.
- **Precondiciones.** Condiciones que han de cumplirse para la realización del caso de uso que se está definiendo.
- **Postcondiciones.** Estado del sistema tras la realización del caso de uso que se está definiendo.
- **Escenario de éxito.** Procedimiento para la consecución del objetivo del caso de uso que se está definiendo.

Se procede a exponer los casos de uso en su descripción tabular de la primera parte del proyecto (de la Tabla 3.22 a la 3.24):

Tabla 3.22: CU - 01.

Identificador	CU - 01
Título	Introducir ruta.
Objetivo	Introducir la ruta en la que se encuentra el fichero <i>XML</i> de coordenadas <i>GPX</i> con extensión <i>.gpx</i> .
Precondiciones	<ul style="list-style-type: none"> ■ El usuario debe disponer del fichero de coordenadas en la máquina. ■ La interfaz de usuario de la aplicación debe estar mostrada.
Postcondiciones	<ul style="list-style-type: none"> ■ La ruta del archivo queda seleccionada para la ejecución de la aplicación.
Escenario de éxito	<ul style="list-style-type: none"> ■ Ejecutar la aplicación. ■ Abrir el menú para seleccionar la ruta del archivo de coordenadas <i>GPX</i>. ■ Navegar entre los directorios del sistema. ■ Seleccionar la ruta del archivo de coordenadas <i>GPX</i> que se desea introducir.

Tabla 3.23: CU - 02.

Identificador	CU - 02
Título	Seleccionar ruta de destino.
Objetivo	Seleccionar la ruta de destino del fichero en el lenguaje del simulador <i>SDL</i> con extensión <i>.evt</i> .
Precondiciones	<ul style="list-style-type: none"> ■ La interfaz de usuario de la aplicación debe estar mostrada.
Postcondiciones	<ul style="list-style-type: none"> ■ La ruta de destino para el fichero <i>SDL</i> queda seleccionada para la ejecución de la aplicación.
Escenario de éxito	<ul style="list-style-type: none"> ■ Ejecutar la aplicación. ■ Abrir el menú para seleccionar la ruta de destino del fichero <i>SDL</i>. ■ Navegar entre los directorios del sistema. ■ Seleccionar la ruta de destino deseada.

Tabla 3.24: CU - 03.

Identificador	CU - 03
Título	Realizar la transformación.
Objetivo	Realizar la transformación del archivo de coordenadas <i>GPX</i> a lenguaje <i>SDL</i> del simulador.
Precondiciones	<ul style="list-style-type: none"> ■ La ruta del fichero de coordenadas <i>GPX</i> con extensión <i>.gpx</i> ya debe estar introducida. ■ La ruta de destino para el fichero en lenguaje <i>SDL</i> debe estar especificada.
Postcondiciones	<ul style="list-style-type: none"> ■ El archivo <i>SDL</i> se genera en la ruta de destino especificada.
Escenario de éxito	<ul style="list-style-type: none"> ■ Una vez seleccionadas las rutas de los ficheros, iniciar el proceso. ■ Aceptar la notificación de que el proceso ha finalizado con éxito. ■ Observar que el fichero se ha creado en la ruta de destino especificada.

3.2.2. Descripción gráfica de casos de uso

En este apartado se presentan los casos de uso en su descripción gráfica. No aparecen a tanto nivel de detalle porque ya se han descrito en profundidad en el apartado anterior. Se muestra la descripción gráfica de los casos de uso en la Figura 3.1.

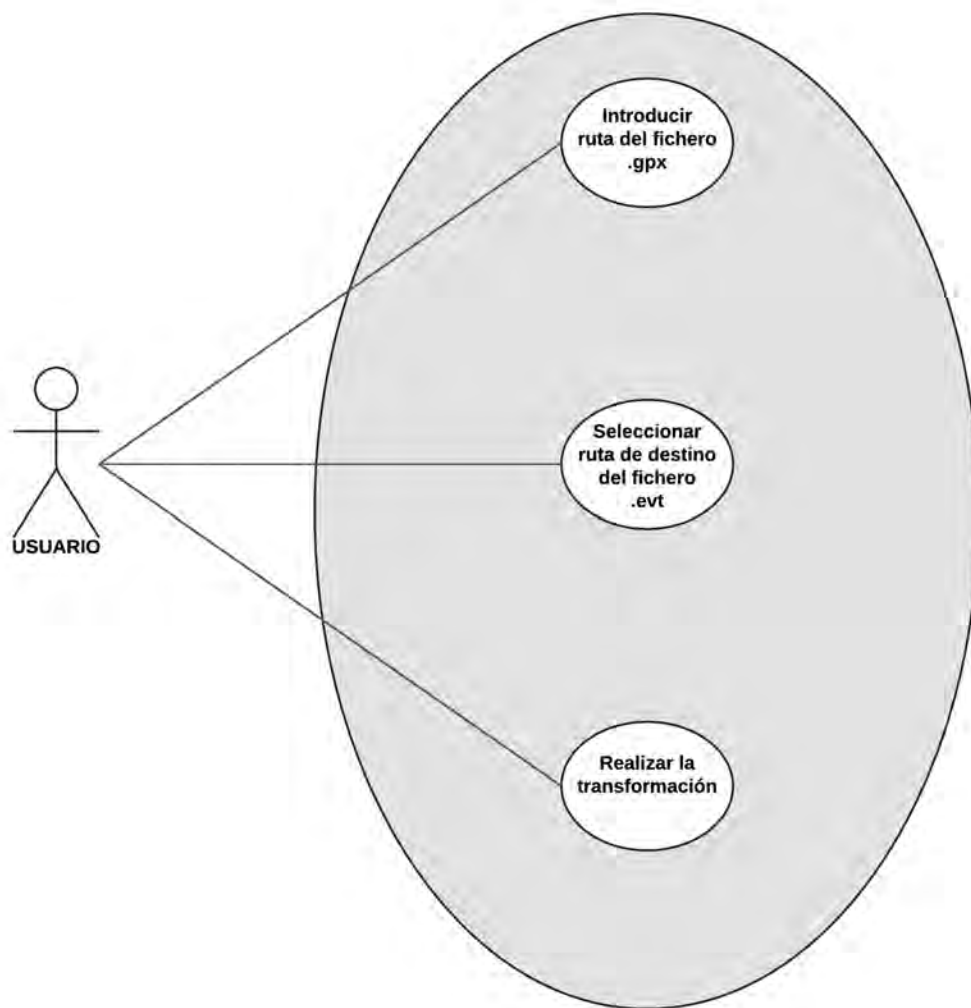


Figura 3.1: Diagrama de casos de uso - gestión de rutas.

3.3. Especificación de requisitos - Toma de decisiones

Los requisitos de esta segunda parte del proyecto se obtienen de diferentes fuentes. Los requisitos más básicos del sistema de esta segunda parte se obtienen directamente del grupo de investigación. Sin embargo, ciertos requisitos han sido extraídos de la opinión recogida en los resultados de encuestas realizadas por el alumno que realiza en proyecto [Anexo II]. En estas encuestas se pregunta acerca de situaciones de peligro en conducción por ciudad a fin de identificarlas como situaciones de estudio en el proyecto. Además, también se pregunta acerca de la aceptación por parte del encuestado de un posible ADAS que ayude al conductor en esas situaciones. Las líneas del proyecto se han guiado en parte por las respuestas a las encuestas.

Para realizar la especificación de los requisitos del sistema de toma de decisiones se emplea una tabla para cada uno de ellos con el mismo formato que en la especificación de requisitos de la primera parte del proyecto.

3.3.1. Requisitos funcionales

Los requisitos funcionales de esta segunda parte del proyecto se presentan a continuación (de la tabla 3.25 a la 3.37):

Tabla 3.25: RF - 01.

Identificador	RF - 01		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo los datos referidos a la dinámica del vehículo: velocidad, acelerador, freno, embrague, marcha, revoluciones por minuto, ángulo de giro del volante, aceleración longitudinal y aceleración lateral.		

Tabla 3.26: RF - 02.

Identificador	RF - 02		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo la información del coche situado detrás del vehículo.		

Tabla 3.27: RF - 03.

Identificador	RF - 03		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo la información del coche situado en diagonal delante y a la derecha del vehículo.		

Tabla 3.28: RF - 04.

Identificador	RF - 04		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo la información del coche situado en diagonal delante y a la izquierda del vehículo.		

Tabla 3.29: RF - 05.

Identificador	RF - 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo la información del coche situado en diagonal detrás y a la derecha del vehículo.		

Tabla 3.30: RF - 06.

Identificador	RF - 06		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se ampliará la ontología del sistema existente añadiendo la información del coche situado en diagonal detrás y a la izquierda del vehículo.		

Tabla 3.31: RF - 07.

Identificador	RF - 07		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema detectará situaciones de peligro y lanzará alarmas ante éstas.		

Tabla 3.32: RF - 08.

Identificador	RF - 08		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema producirá como salida una alarma cuando haya un coche delante a poca o muy poca distancia, el conductor no esté atento y la velocidad del vehículo sea media o alta.		

Tabla 3.33: RF - 09.

Identificador	RF - 09		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema producirá como salida una alarma cuando haya peatones cerca o muy cerca del vehículo, que estén cruzando y cuando el conductor no esté mirando al frente del vehículo.		

Tabla 3.34: RF - 10.

Identificador	RF - 10		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema producirá como salida una alarma cuando se vaya a adelantar a un vehículo delante que esté parado o en movimiento y cerca o muy cerca, habiendo un vehículo detrás en movimiento cerca o muy cerca.		

Tabla 3.35: RF - 11.

Identificador	RF - 11		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema producirá como salida una alarma cuando haya un coche delante o delante a la derecha aparcado que pueda querer arrancar y salir mientras el conductor no tenga la vista hacia el frente.		

Tabla 3.36: RF - 12.

Identificador	RF - 12		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Descripción	El sistema producirá como salida una alarma cuando haya un coche aparcado delante del vehículo y haya un peatón justo detrás que vaya a cruzar la carretera.		

Tabla 3.37: RF - 13.

Identificador	RF - 13		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Entre las alarmas que tiene como salida el sistema habrá una jerarquía puesto que solo se puede tener una salida.		

3.3.2. Requisitos no funcionales

Los requisitos no funcionales de esta segunda parte del proyecto se presentan a continuación (de la tabla 3.38 a la 3.42):

Tabla 3.38: RNF - 01.

Identificador	RNF - 01		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La ontología será implementada en el lenguaje <i>RDF</i> .		

Tabla 3.39: RNF - 02.

Identificador	RNF - 02		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Se trabajará la ontología mediante el programa TopBraid Composer.		

Tabla 3.40: RNF - 03.

Identificador	RNF - 03		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema de reglas será implementado en el lenguaje de programación JAVA.		

Tabla 3.41: RNF - 04.

Identificador	RNF - 04		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Las consultas sobre instancias de la ontología serán implementadas en lenguaje <i>SPARQL</i> .		

Tabla 3.42: RNF - 05.

Identificador	RNF - 05		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema de reglas se integrará en los sistemas de agentes que estén implementados o se vayan a implementar.		

3.4. Casos de uso - Toma de decisiones

En este apartado de la memoria se definen los casos de uso de la segunda parte del proyecto: el sistema de toma de decisiones basado en reglas. De nuevo, se describirán de manera tabular y de manera gráfica.

3.4.1. Descripción tabular de casos de uso

Para especificar los casos de uso en su forma tabular, se usa el formato de las tablas que se utilizaron en la descripción de los casos de uso de la primera parte del proyecto.

En este sistema, el usuario no realiza las acciones directamente, sino que interactúa con el sistema cuando se producen determinadas situaciones. Por lo tanto, el usuario conduce en el simulador de conducción hasta que se da un escenario específico, y es entonces cuando se consiguen los objetivos con el sistema.

Se procede a exponer los casos de uso en su descripción tabular de la segunda parte del proyecto (de la tabla 3.43 a la 3.48):

Tabla 3.43: CU - 01.

Identificador	CU - 01
Título	Evitar colisión frontal.
Objetivo	Evitar la colisión frontal con el vehículo de delante.
Precondiciones	<ul style="list-style-type: none"> ■ Existe un coche delante del vehículo. ■ El conductor no está mirando al frente. ■ La distancia al vehículo delantero es baja o muy baja. ■ La velocidad a la que va el conductor es media o alta.
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que existe un vehículo no visualizado cercano de frente.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de colisión con el vehículo delantero. ■ El sistema lanza la alarma correspondiente a la existencia de un vehículo no visualizado cercano de frente. ■ El conductor reacciona evitando la colisión frontal.

Tabla 3.44: CU - 02.

Identificador	CU - 02
Título	Evitar atropello de peatón no visualizado.
Objetivo	Evitar atropellar a peatones no visualizados que crucen la carretera.
Precondiciones	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Existen peatones delante del vehículo o en diagonal hacia la derecha. ■ El peatón sigue una trayectoria de derecha a izquierda. ■ El conductor no está mirando al frente. ■ La distancia al peatón es baja o muy baja. ■ La velocidad a la que va el conductor es media o alta.
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que existe un peatón no visualizado que cruza.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de arrollamiento a un peatón no visualizado. ■ El sistema lanza la alarma correspondiente a la existencia de un peatón no visualizado que cruza. ■ El conductor reacciona evitando atropellar al peatón.

Tabla 3.45: CU - 03.

Identificador	CU - 03
Título	Evitar atropello de peatón no visualizado.
Objetivo	Evitar atropellar a peatones no visualizados que crucen la carretera.
Precondiciones	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Existen peatones delante del vehículo o en diagonal hacia la izquierda. ■ El peatón sigue una trayectoria de izquierda a derecha. ■ El conductor no está mirando al frente. ■ La distancia al peatón es baja o muy baja. ■ La velocidad a la que va el conductor es media o alta.
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que existe un peatón no visualizado que cruza.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de arrollamiento a un peatón no visualizado. ■ El sistema lanza la alarma correspondiente a la existencia de un peatón no visualizado que cruza. ■ El conductor reacciona evitando atropellar al peatón.

Tabla 3.46: CU - 04.

Identificador	CU - 04
Título	Evitar colisión trasera en el adelantamiento.
Objetivo	Evitar colisiones con el coche de atrás cuando ambos, conductor y vehículo trasero, quieren adelantar.
Precondiciones	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Existe un vehículo delante cerca o muy cerca, ya sea parado o en movimiento. ■ Existe un vehículo detrás cerca o muy cerca en movimiento. ■ El ángulo de giro del volante indica que el volante ha sido girado a la izquierda
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que hay que prestar atención al coche de atrás al realizar el adelantamiento.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de colisión con el coche de atrás al realizar un adelantamiento. ■ El sistema lanza la alarma correspondiente al peligro de colisión con el coche de atrás. ■ El conductor reacciona evitando atropellar al peatón.

Tabla 3.47: CU - 05.

Identificador	CU - 05
Título	Evitar colisión con coche detenido que arranca.
Objetivo	Evitar la colisión con un coche detenido delante del vehículo que sale cuando el conductor no está mirando al frente.
Precondiciones	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Existe un vehículo enfrente o en diagonal a la derecha, parado, y está a una distancia media, baja o muy baja. ■ El conductor no está mirando al frente
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que hay que hay un coche detenido que puede querer salir.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de que un coche detenido delante del vehículo pueda salir mientras el conductor no está atento. ■ El sistema lanza la alarma correspondiente al peligro de salida de un coche mientras el conductor no está atento. ■ El conductor reacciona evitando la colisión con el vehículo que sale.

Tabla 3.48: CU - 06.

Identificador	CU - 06
Título	Evitar atropello de peatón no visible.
Objetivo	Evitar atropellar a peatones que cruzan y no visibles, es decir, aparecen de detrás de algún objeto que los oculta.
Precondiciones	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Existe un vehículo en diagonal hacia delante obstaculiza la visión de los peatones. ■ Existen peatones que se encuentren detrás del vehículo, ocultos, y su trayectoria indica que va a cruzar la carretera.
Postcondiciones	<ul style="list-style-type: none"> ■ Se lanza la alarma de que existe peatón no visible que cruza.
Escenario de éxito	<ul style="list-style-type: none"> ■ El usuario conduce en el simulador de conducción. ■ Se da la situación que implica el peligro de que un peatón no visible puede cruzar. ■ El sistema lanza la alarma correspondiente al peligro que cruce el peatón que no es visible. ■ El conductor reacciona evitando atropellar al peatón.

3.4.2. Descripción gráfica de casos de uso

En este apartado se presenta la definición gráfica de los casos de uso. En esta representación se refleja el hecho de que el usuario no interacciona directamente con el sistema, sino que se dedica a la conducción hasta que se producen situaciones específicas donde sí interviene el sistema (ver Figura 3.2).

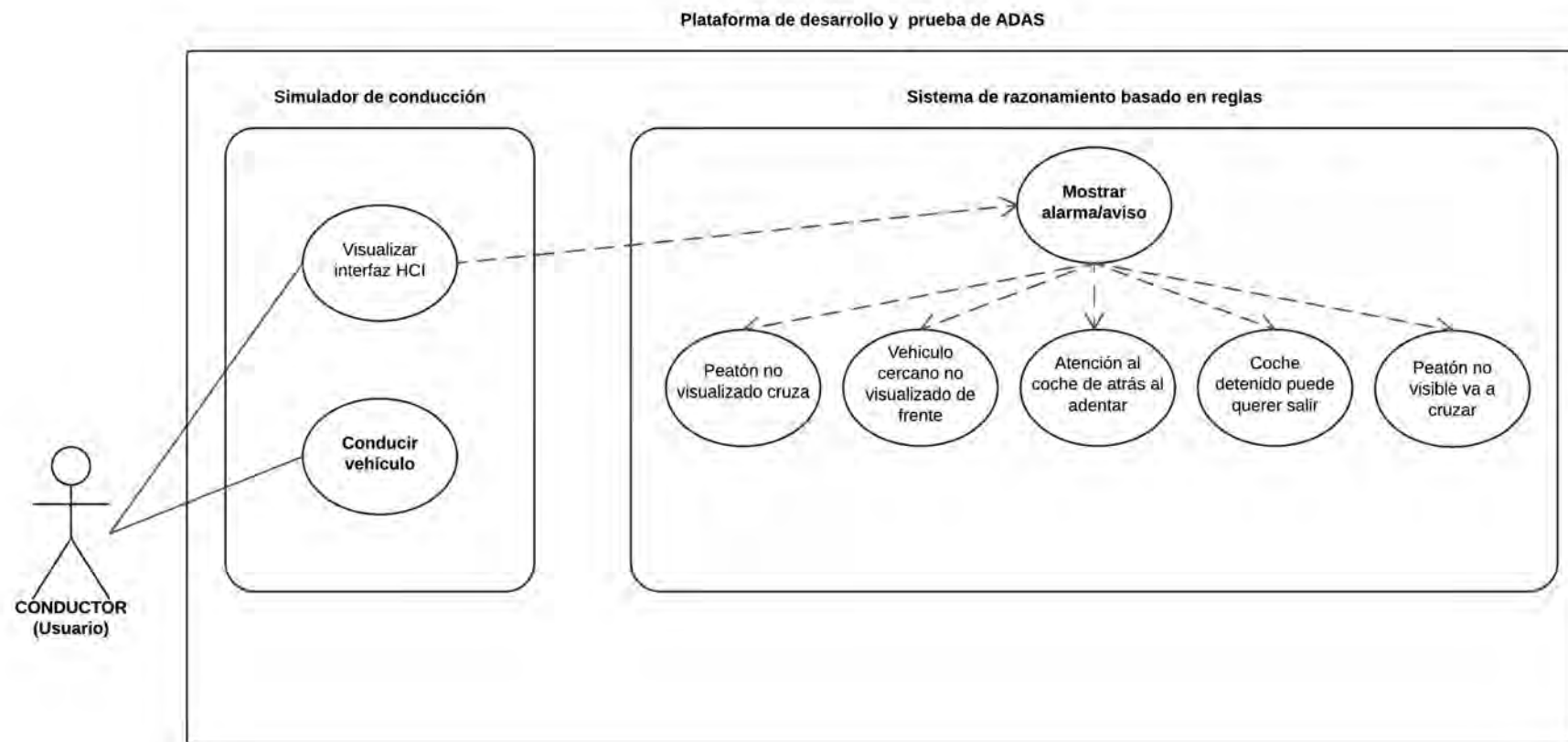


Figura 3.2: Diagrama de casos de uso - toma de decisiones.

Capítulo 4

Arquitectura y diseño del sistema

En este capítulo de la memoria se describe el proceso del diseño de los sistemas a desarrollar. El objetivo de este proceso es definir la arquitectura de cada sistema y el entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema.

Por lo tanto, se describirán en las siguientes secciones los componentes o módulos en los que se dividen los sistemas a implementar, así como la interacción entre ellos.

4.1. Gestión de rutas

Para el sistema implementado en la primera parte del proyecto, se distinguen cuatro módulos principales: el módulo de interfaz gráfica, el módulo de entrada-salida, el módulo de transformación de coordenadas y el módulo de tratamiento de coordenadas *UTM* (*Universal Transversal de Mercator*). Se muestra en la Figura 4.1 un diagrama general de la arquitectura del sistema descompuesta en módulos, indicando la interacción entre cada uno de ellos.

Un diseño alternativo a éste que se plantea podría haber sido el resultado de combinar los dos últimos módulos, transformación de coordenadas y tratamiento de coordenadas *UTM*, en uno solo. Esto se debería a que el módulo de transformación de coordenadas solo tiene una función dentro del sistema, que es la transformación de coordenadas del sistema *GPX* al sistema *UTM*, y podría unirse con el módulo de tratamiento de coordenadas *UTM* para evitar un módulo con una funcionalidad única. De esta manera, se tendría una división en menos componentes. Sin embargo, se decide emplear el diseño que separa ambos módulos para hacer la arquitectura más modular y, con ello, se contribuye a la reutilización de software y se mejora el rendimiento humano pudiendo realizarse programación en equipo (aunque en este caso no procede). Además, se elige este diseño con vista a una posible ampliación del sistema, en la que, por ejemplo, puedan hacer falta transformaciones entre otros sistemas de coordenadas.

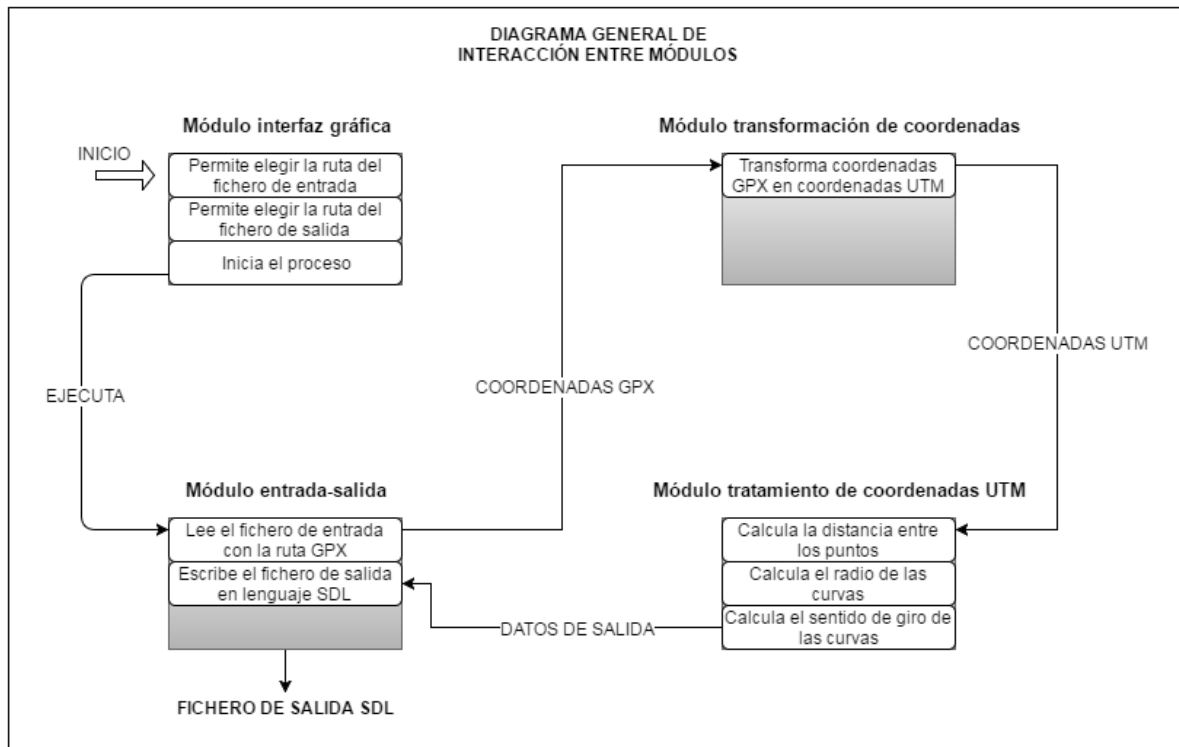


Figura 4.1: Arquitectura del sistema - Gestión de rutas.

En los siguientes apartados se describe, en detalle, cada uno de los módulos que componen el sistema de gestión de rutas.

4.1.1. Módulo interfaz gráfica

El primer módulo del sistema es el módulo interfaz gráfica. Este módulo es el que permite la comunicación entre la aplicación y el usuario de una manera sencilla y eficaz.

Mediante una interfaz gráfica simple, el usuario será capaz de seleccionar la ruta donde se ubica el fichero de coordenadas *GPX* con extensión *.gpx*. De la misma manera, podrá escoger la ruta de destino para el fichero de salida de la aplicación en lenguaje *SDL* con extensión *.etv* e iniciar el proceso de ejecución de la aplicación. Cuando la ejecución acabe, con éxito o no, se notificará mediante un mensaje en una ventana emergente.

4.1.2. Módulo entrada-salida

El segundo módulo del sistema es el módulo entrada-salida. Este módulo es el encargado de leer el fichero de entrada con la ruta *GPX* cuya ruta se ha indicado en el módulo descrito previamente. Después de leer el fichero de entrada, éste se almacena para ser utilizado por los módulos de transformación de coordenadas y de tratamiento de coordenadas *UTM*.

Una vez realizados todos los cálculos de los módulos mencionados anteriormente, se escribe el fichero de salida SDL con los resultados. Este fichero se ubicará en la ruta indicada como parámetro en el módulo interfaz gráfica.

4.1.3. Módulo transformación de coordenadas

El tercer módulo del sistema es el módulo transformación de coordenadas. El objetivo de este módulo es la transformación de las coordenadas de un sistema a otro. En este proyecto, este módulo solo convierte coordenadas que están en el sistema *GPX* en coordenadas en el sistema *UTM*, para posteriormente realizar cálculos con ellas. Sin embargo, este módulo podría ampliarse, según la necesidad, con más transformaciones entre sistemas de coordenadas.

Una vez realizada la transformación, se almacenan las nuevas coordenadas para ser utilizadas por el módulo tratamiento de coordenadas *UTM*.

4.1.4. Módulo tratamiento de coordenadas *UTM*

El cuarto y último módulo del sistema es el módulo tratamiento de coordenadas *UTM*. Una vez se dispone de las coordenadas en el sistema *UTM*, este módulo se encarga de analizarlas, procesarlas y calcular lo necesario para obtener la ruta en el lenguaje SDL del simulador. Este proceso se divide en tres subprocesos: el cálculo de las distancias entre cada par de puntos, el cálculo del radio de las curvas formadas por cada tres puntos y el cálculo del sentido de giro de dichas curvas.

Una vez se dispone de toda esa información, ya se puede escribir el fichero en lenguaje SDL del simulador con extensión .evt. El encargado de este proceso es, como se ha indicado en el apartado correspondiente, el módulo entrada-salida.

4.2. Toma de decisiones

El sistema de toma de decisiones basado en reglas implementado en este proyecto no es independiente, como el sistema de gestión de rutas, sino que está incluido dentro de un sistema multiagente ya implementado por el grupo de investigación CAOS de la Universidad Carlos III de Madrid [3]. Por lo tanto, conviene ubicar primero el sistema de toma de decisiones dentro del sistema multiagente implementado.

En la figura 4.2 se presenta el diagrama de componentes que describe la arquitectura completa del sistema multiagente ya implementado para el simulador.

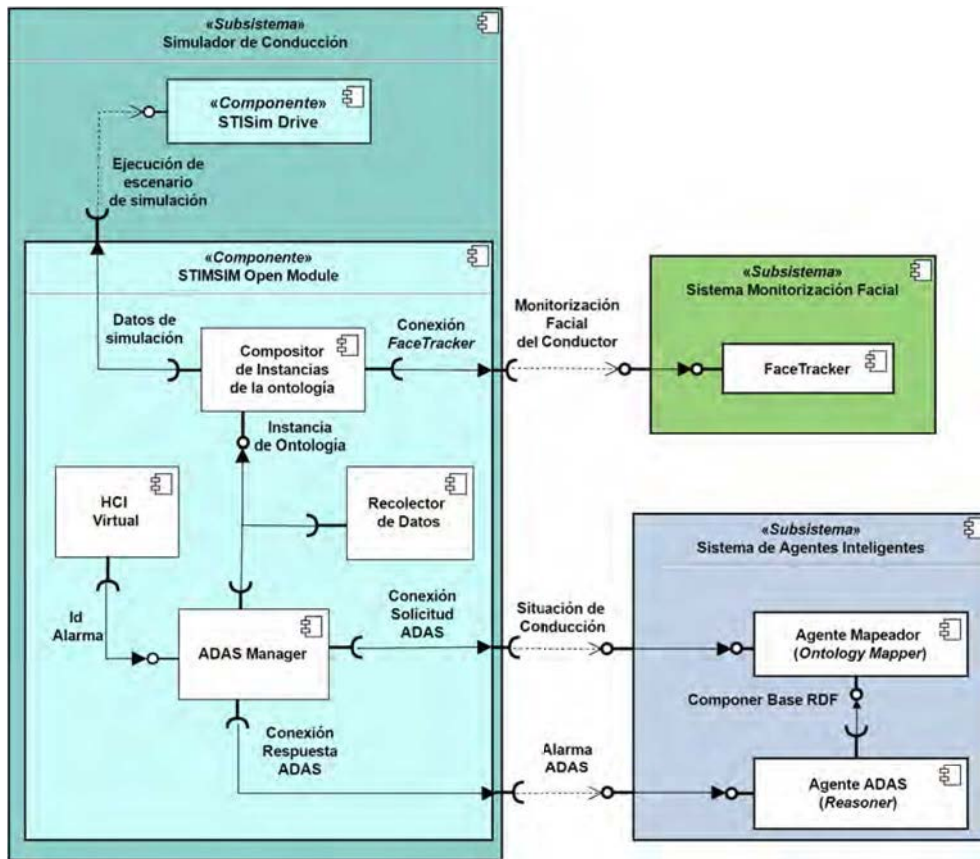


Figura 4.2: Arquitectura del sistema multiagente del simulador.

Como se observa en la Figura 4.2, el sistema se divide principalmente en 4 grandes componentes, que son:

- **Componente Simulador de Conducción.** Corresponde con el software de simulación ejecutado en el sistema de simulación de conducción.
- **Componente STISIM Open Module.** Corresponde con el módulo de desarrollo de *STISIM Drive* en el que serán implementadas todas las funcionalidades requeridas por el sistema para la obtención de información.
- **Componente Sistema de Monitorización Facial.** corresponde con el equipo hardware encargado de la ejecución del software de monitorización facial del conductor.
- **Componente Sistema de Agentes Inteligentes.** Este último componente corresponde con el equipo hardware encargado de la ejecución de la plataforma de agentes inteligentes. Aquí es donde se encuentra el sistema de toma de decisiones basado en reglas desarrollado en este proyecto, más en concreto, dentro del subcomponente *Agente ADAS (Reasoner)*.

Para el sistema implementado en la segunda parte del proyecto, se distinguen tres módulos principales: el módulo de gestión de consultas, el módulo ontología y el de módulo razonamiento. Se muestra en la Figura 4.3 un diagrama general de la arquitectura del sistema descompuesta en módulos indicando la interacción entre cada uno de ellos.

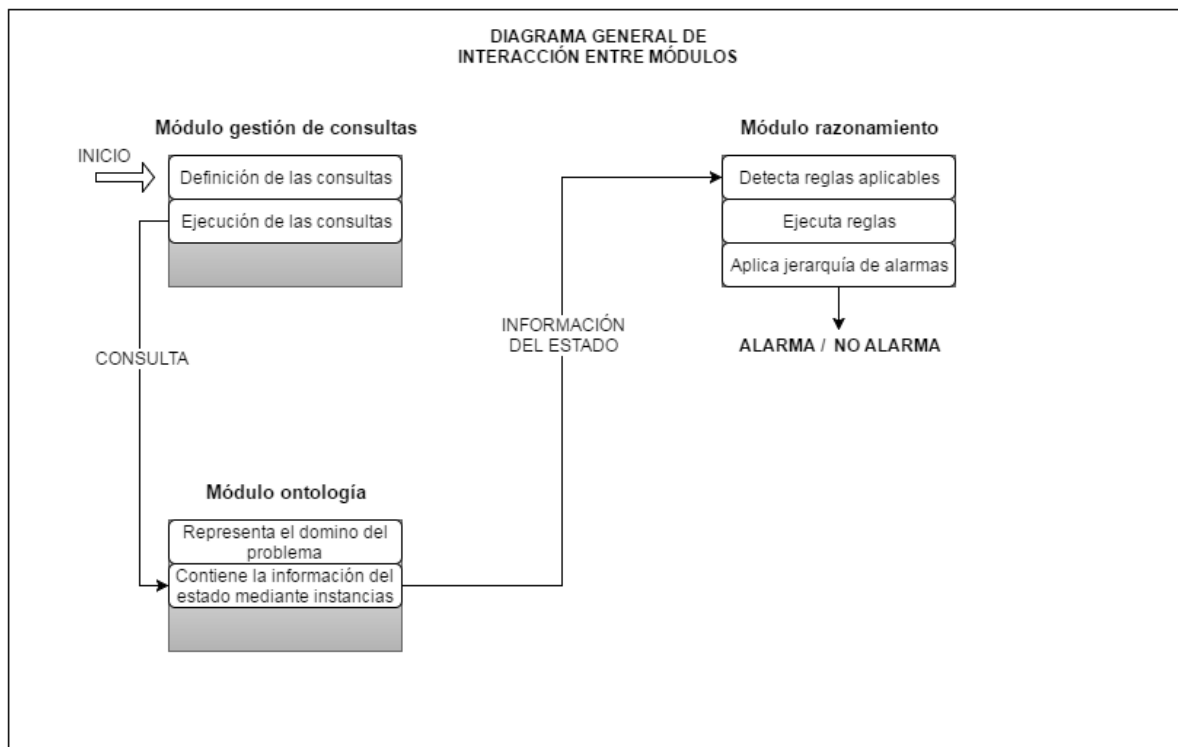


Figura 4.3: Arquitectura del sistema - Toma de decisiones.

Como se observa en el diagrama de la Figura 4.3, se trata de un diseño sencillo con funcionalidades claramente separadas entre sí. Un diseño alternativo al escogido habría sido el resultado de separar la funcionalidad de la jerarquía de alarmas en un módulo aparte. Sin embargo, esta separación no tendría utilidad con vistas a una ampliación, ya que la jerarquía entre las alarmas está fuertemente ligada a las reglas, y la modificación de unas conllevaría la modificación de la otra. Por lo tanto, se opta por un diseño más sencillo y robusto, con alta cohesión y bajo acoplamiento, es decir, con pocas dependencias entre módulos.

En los siguientes apartados se describe uno a uno cada uno de los módulos que componen el sistema de toma de decisiones basado en reglas.

4.2.1. Módulo gestión de consultas

El primer módulo del sistema de toma de decisiones es el módulo gestión de consultas. Este módulo es el encargado de la definición y la ejecución de las consultas necesarias para la posterior formulación de las reglas. Estas consultas reunirán los datos sobre el entorno del vehículo, así como del estado del conductor, y los almacenarán para su uso posterior.

Las consultas se definen en base a una ontología, ya definida inicialmente por el grupo de investigación CAOS de la Universidad Carlos III de Madrid, y ampliada en el presente proyecto. Dicha ontología se recoge en el módulo descrito en el subapartado 4.2.2

4.2.2. Módulo ontología

El segundo módulo del sistema de toma de decisiones es el módulo ontología. Este módulo representa el dominio del problema, y sus objetivos principales son limitar la complejidad y organizar la información de forma correcta. De esta manera, este módulo contiene la información del estado, tanto del entorno del vehículo como del conductor, en cada momento.

Este módulo es, pues, consultado por el módulo de gestión de consultas para proporcionar la información necesaria en cada momento al módulo de razonamiento, el cual se describe en el subapartado 4.2.3

4.2.3. Módulo razonamiento

El tercer y último módulo del sistema de toma de decisiones es el módulo de razonamiento. Este módulo tiene como objetivo realizar todas las tareas de razonamiento con los datos que se han obtenido mediante consultas del sistema. En él, se encuentran implementadas todas las reglas que conforman la base de reglas que forman parte del sistema de toma de decisiones. De estas reglas, cada vez que se produzca una petición al sistema, se identifican las reglas que se pueden aplicar y se ejecutan, produciendo así la alarma correspondiente.

Sin embargo, puede darse el caso de que varias reglas se activen para una misma situación. Dado que el sistema actualmente solo puede tener una salida, es necesario incluir una jerarquía entre alarmas producidas por las reglas. La jerarquía de alarmas también forma parte de este módulo, y está diseñada de tal manera que se tiene en cuenta la regla que se active bajo la situación que más peligro represente.

Por lo tanto, tras realizarse el proceso de razonamiento dada la situación en cada petición, se muestra como salida del sistema la alarma disparada por la regla más relevante dentro de la jerarquía.

Capítulo 5

Implementación del sistema

En este capítulo de la memoria se describe el proceso de desarrollo de los sistemas implementados, describiendo en primer lugar el desarrollo del sistema de gestión de rutas y finalizando con el desarrollo del sistema de toma de decisiones basado en reglas. Dentro de cada sistema, tomando como punto de partida el modelo descrito en la etapa de diseño (Capítulo 4), se procede a implementar los diseños especificados en el modelo de diseño.

5.1. Gestión de rutas

Dentro de esta sección se describirá el proceso de implementación del primero de los dos sistemas del trabajo de fin de grado, el sistema de gestión de rutas. Siguiendo el esquema del diseño, se describirá por módulos, comenzando por el módulo entrada-salida, y siguiendo por el módulo de transformación de coordenadas, el módulo de tratamiento de coordenadas *UTM*, y finalizando con el módulo interfaz gráfica.

El lenguaje de programación escogido para el desarrollo del sistema de gestión de rutas ha sido *JAVA*, ya que es un lenguaje robusto, dinámico y simple que ofrece una gran portabilidad en las aplicaciones, es decir, que el mismo código funcionará en diferentes máquinas sin necesidad de ser recompilado.

Para este subproyecto, existen únicamente dos clases: la clase principal donde se encuentran los métodos implementados para los módulos entrada-salida, transformación de coordenadas y tratamiento de coordenadas *UTM*, y una clase para la creación de la interfaz gráfica, que será ejecutada desde la clase principal.

5.1.1. Entrada-Salida

El objetivo principal del sistema de gestión de rutas es convertir un fichero de ruta *GPX* generado externamente, es decir, con cualquier herramienta web de las descritas en la sección 1.6, en un fichero con extensión *.evt* escrito en el lenguaje de definición de escenarios del simulador utilizado, *SDL* (Scenario Development Language). Por lo tanto, lo primero que se debe implementar es un método que sea capaz de leer el fichero de coordenadas *GPX* dado, el cual sigue una estructura mediante etiquetas *XML* bastante clara.

El método implementado lee archivos de este tipo (ejemplo en sección 1.3). Dicho método se encuentra descrito en el Algoritmo 1.

Algoritmo 1 Algoritmo para la lectura del fichero de coordenadas GPX

Entrada: Archivo de ruta GPX

Salida: Si el archivo de ruta tiene el formato correcto, almacena los valores de las coordenadas GPX para su uso posterior.

```
1: mientras Archivo no vacío hacer
2:   Se ignoran todos los caracteres hasta encontrar un espacio en blanco, y se lee a
     partir de ahí
3:   si Si los tres primeros caracteres forman la cadena “lat” entonces
4:     Se guarda el valor de la latitud indicado a continuación
5:     si Si los tres siguientes caracteres forman la cadena “lon” entonces
6:       Se guarda el valor de la longitud indicado a continuación
7:     si no
8:       Se vuelve a 1
9:     fin si
10:  si no
11:    Se vuelve a 1
12:  fin si
13: fin mientras
```

De esta manera, al finalizar la ejecución del método, se tendrán almacenados todos los valores de latitud y longitud de los diferentes puntos que componen la ruta pasada como argumento de entrada, permitiendo así que sean utilizados por el resto del programa. Una solución alternativa a la implementación de este método podría haber sido navegando a través de él leyendo etiquetas.

Una vez procesados todos los datos, el último paso de la ejecución total es la escritura del fichero de salida. Por lo tanto, en este módulo se implementa también el método correspondiente a la escritura del fichero en lenguaje *SDL* del simulador con extensión *.evt* (explicado en la sección 1.3).

El método implementado consiste, por lo tanto, en crear un fichero con extensión *.evt* y escribir en él cada uno de los eventos identificados en la ejecución del programa de manera acorde a la definición de los mismos. Cabe destacar que la primera línea del fichero debe ser “METRIC”, para poder utilizar las medidas del sistema métrico internacional, y que la última línea debe indicar la distancia máxima que se va a recorrer y el fin de la simulación con el comando “End Simulation”.

5.1.2. Transformación de coordenadas

Una vez leído el fichero y obtenida la ruta en sus coordenadas de latitud y longitud, se procede a transformar las coordenadas del sistema geodésico al sistema *UTM*. Esta transformación es necesaria debido a que las magnitudes en el sistema *UTM* se expresan en metros al nivel del mar, que es la base de la proyección del elipsoide de referencia. Esto hace que trabajar con ellas sea más sencillo, puesto que las distancias euclídeas entre puntos vendrán dadas en metros directamente.

El método implementado transforma todas las coordenadas de la ruta al sistema de coordenadas *UTM*, y las almacena para su uso posterior. Para transformar las coordenadas del sistema geodésico al sistema *UTM*, se emplean las ecuaciones de *Coticchia-Surace*, que son el método más empleado para la resolución del problema de la transformación de las coordenadas entre estos sistemas. El proceso completo seguido y codificado se describe en [34].

5.1.3. Tratamiento de coordenadas *UTM*

Una vez transformadas todas las coordenadas al sistema *UTM*, se pueden realizar los cálculos con ellas para obtener la información necesaria. Siguiendo el diseño del sistema descrito en la sección anterior (sección 4) de la memoria, se desarrollan tres métodos para procesar las coordenadas *UTM* y obtener los datos relevantes para la escritura del fichero final en lenguaje *SDL*.

El primero de los tres métodos tiene como objetivo calcular la distancia entre cada par de puntos. Se trata de un método sencillo que, para cada par de puntos, calcula la distancia euclídea y la almacena para su uso posterior.

El segundo método tiene como objetivo el cálculo del radio de las curvas que haya en el trazado de la ruta. Para ello, el método calculará en primer lugar dos vectores en cada curva, desde el punto inicial al punto intermedio de la curva (u), y desde el punto final hasta, también, el punto intermedio de la misma (v), y hallará sus vectores perpendiculares (up , vp). Con esos vectores perpendiculares, calcula sus rectas correspondientes (ru , rv) y calcula el punto de corte entre ellas resolviendo el sistema de ecuaciones que plantean. De esta manera, desde el punto de corte hasta el punto intermedio de cada curva, se obtiene el valor del radio de la curva en cuestión. En la Figura 5.1 se presenta un esquema del proceso.

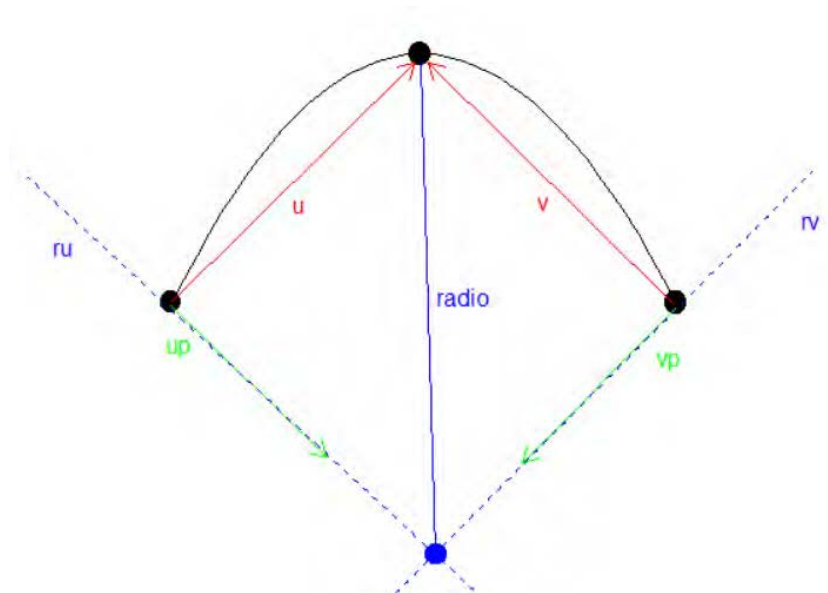


Figura 5.1: Cálculo del radio de la curva.

El tercer y último método de este módulo de tratamiento de coordenadas tiene como

objetivo calcular el sentido de giro de dichas curvas. Para ello, se vuelven a calcular dos vectores. Sin embargo, esta vez los vectores serán desde el punto inicial al punto intermedio de la curva (v) y desde el punto inicial hasta el punto final (u).

Teniendo esos dos vectores, se calcula el ángulo β que forman de la siguiente manera, siendo el vector u de referencia el formado por los puntos inicial y final, y siendo el vector v el formado por los puntos inicial e intermedio de la curva:

$$\arctan(v_x u_y - v_y u_x, v_x u_x + v_y u_y)$$

Esta forma de calcular el ángulo entre dos vectores devuelve un ángulo en el intervalo $[-\pi, \pi]$ en radianes. Entonces podemos determinar que:

- Si el valor devuelto es 0, quiere decir que no se trata de una curva, sino de una recta (ver Figura 5.2).

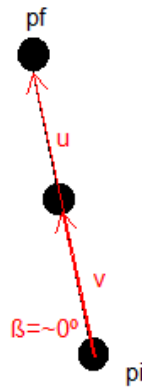


Figura 5.2: Caso de ángulo 0° .

- Si el valor devuelto es positivo, el sentido de giro de la curva es hacia la izquierda. Esto se debe a que, en este caso, el vector u formado a partir de los puntos inicial y final está a la izquierda del vector v , formado por los puntos inicial e intermedio, por lo que el giro de la curva sería a izquierdas (ver Figura 5.3).

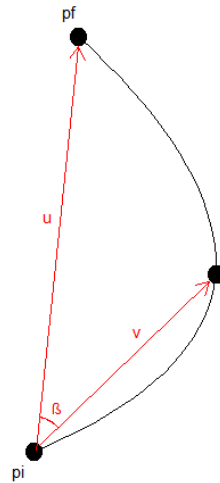


Figura 5.3: Caso de ángulo positivo.

- Si el valor devuelto es negativo, el sentido de giro de la curva es hacia la derecha. Esto se debe a que, en este caso, el vector u formado a partir de los puntos inicial y final está a la derecha del vector v , formado por los puntos inicial e intermedio, por lo que el giro de la curva sería a derechas (ver Figura 5.4).

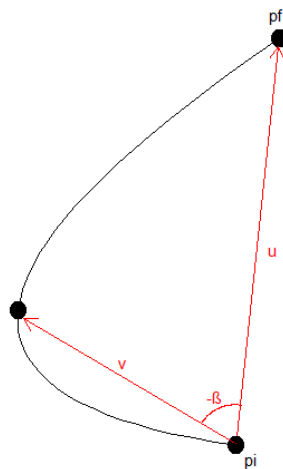


Figura 5.4: Caso de ángulo negativo.

Una solución alternativa a este último método sería emplear ecuaciones y metodologías de topografía, pero, por la complejidad que implica, es preferible el método descrito, que mantiene una solución sencilla, eficiente, y fácil de implementar.

5.1.4. Interfaz gráfica

Por último, se describe la interfaz gráfica del sistema, que es implementada en una clase aparte e invocada desde la clase principal, como se ha indicado anteriormente. Esta interfaz, como se ha especificado en el diseño del sistema, permite al usuario elegir la ruta del fichero de entrada, elegir la ruta del fichero de salida y permite iniciar el proceso de ejecución.

Se trata de una interfaz sencilla, con dos campos de texto con etiqueta y botón selector de ficheros asociados, y un tercer botón para iniciar la ejecución del sistema. En cada etiqueta se indica qué es lo que se debe introducir en ese campo de texto, y en el botón se puede navegar a través de los ficheros del computador en el que se ejecute y seleccionar directamente la ruta desde dicho navegador. En la Figura 5.5 se muestra una imagen de la interfaz del sistema.

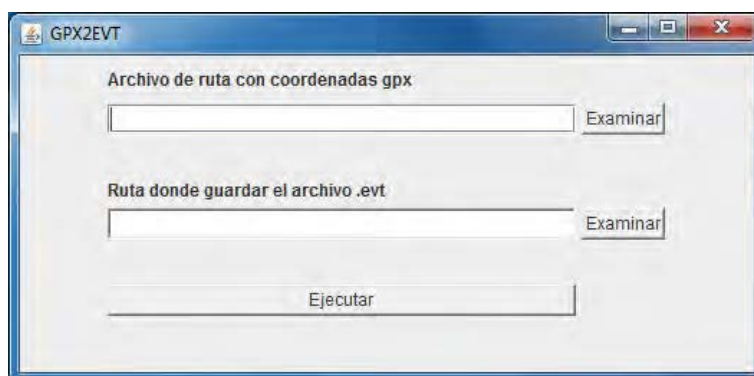


Figura 5.5: Interfaz del sistema - Gestión de rutas.

5.2. Toma de decisiones

Dentro de esta sección se describirá el proceso de implementación del segundo de los dos sistemas del trabajo de fin de grado, el sistema de toma de decisiones basado en reglas. Siguiendo el esquema del diseño (Capítulo 4), se describirá por módulos. Sin embargo se alterará el orden de la siguiente manera: se comienza por el módulo de la ontología, se sigue por el módulo de gestión de consultas, y se finaliza por el módulo de razonamiento. Antes de esta descripción por módulos, se expondrán cuestiones generales.

Este sistema, como se indicó en la arquitectura general (ver Figura 4.2) descrita en el capítulo de la memoria correspondiente al diseño del sistema (Capítulo 4), está dentro de un sistema multiagente construido para trabajar sobre el simulador de conducción. Más concretamente, la implementación del sistema de toma de decisiones basado en reglas forma parte de los agentes de dicho sistema que se dedican a la fusión y al análisis de los datos que proporciona el simulador. A su vez, el sistema multiagente en el que se encuentra el sistema de toma de decisiones, forma parte de un Sistema Avanzado de Asistencia a la Conducción, que es el objetivo principal.

Este sistema multiagente está implementado usando el *framework* de desarrollo de sistemas de agentes *JADE* (*JAVA Agent DEvelopment Framework*) [35]. Éste es un software enteramente implementado en el lenguaje de programación *JAVA*, que simplifica el desarrollo de sistemas multiagentes a través de un *middle-ware* que cumple con las especificaciones *FIPA* [36] y mediante un set de herramientas gráficas que facilitan la depuración y el despliegue de las fases.

Por lo tanto, el desarrollo del sistema multiagente está codificado en lenguaje de programación *JAVA*. Sin embargo, se usan otros lenguajes para la realización de tareas específicas, como por ejemplo, el manejo de la ontología. Estos lenguajes se integran mediante el uso de la librería *Apache Jena* [37], y se exponen en la sección 5.2.1.

El objetivo del sistema de toma de decisiones es decidir, dada una situación de conducción descrita por el estado de la vía, del vehículo y su entorno, y del conductor, si existe un peligro o riesgo de accidente y, en caso afirmativo, informar al conductor mediante un sistema de alarmas. Habrá, por tanto, diferentes tipos de alarmas para diferentes situaciones.

Las situaciones y escenarios objeto de estudio son definidas previamente. Para identificar y definir estos escenarios, se realizó una encuesta a conductores, tanto noveles como con una experiencia de más de diez años con carnet de conducir, en la que se preguntaba acerca de situaciones de peligro en conducción por ciudad y acerca de la aceptación de un Sistema Avanzado de Asistencia a la Conducción que asista al conductor en esos casos [Anexo II].

Tras este estudio y el análisis de los resultados de las encuestas, se definieron las siguientes cinco situaciones:

- **Situación 1 - Riesgo de colisión frontal.** El conductor se encuentra distraído en una carretera en la que existe tráfico, y la distancia al coche que precede al del conductor se recorta llegando a ser muy pequeña (ver Figura 5.6).

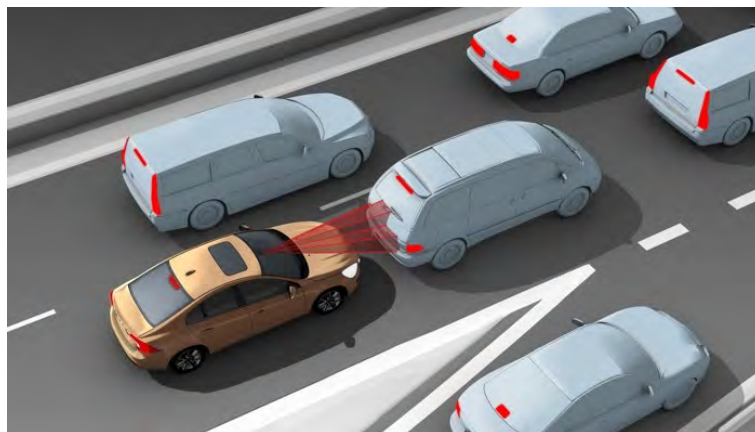


Figura 5.6: Situación 1 - Riesgo de colisión frontal (Fuente: saladepresa.racc.es).

- **Situación 2 - Riesgo de atropello.** Se está realizando una conducción por vía urbana y un peatón cruza la carretera a una distancia relativamente cercana al vehículo (ver Figura 5.7).



Figura 5.7: Situación 2 - Riesgo de atropello (Fuente: www.forotransportistas.es).

- **Situación 3 - Riesgo de colisión trasera.** Se va a realizar un adelantamiento, pero el coche de detrás del vehículo del conductor ya estaba adelantando (ver Figura 5.8).



Figura 5.8: Situación 3 - Riesgo de colisión trasera (Fuente: www.circulaseguro.com).

- **Situación 4 - Riesgo de colisión lateral.** Se está realizando una conducción por vía urbana y un coche que estaba detenido sale sin señalizarlo (ver Figura 5.9).



Figura 5.9: Situación 4 - Riesgo de colisión lateral (Fuente: www.autofacil.es).

- **Situación 5 - Riesgo de atropello de peatón sin visualizar.** Se está realizando una conducción por vía urbana y un peatón cruza la carretera saliendo de detrás de un vehículo aparcado o de un objeto que haga que el peatón no sea visible para el conductor (ver Figura 5.10).

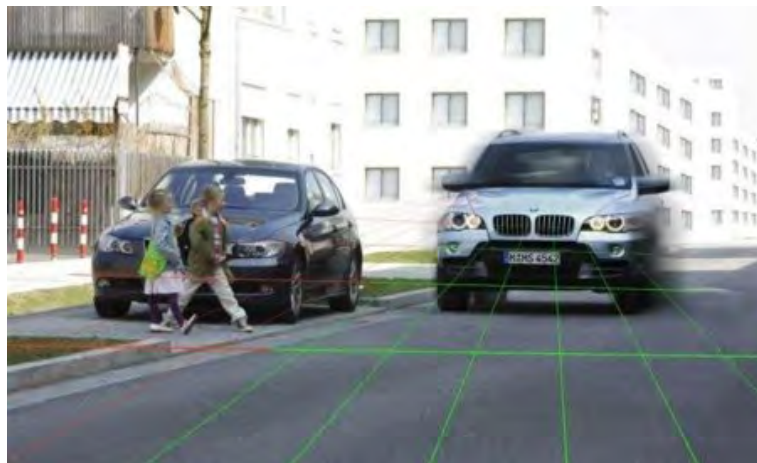


Figura 5.10: Situación 5 - Riesgo de atropello de peatón sin visualizar (Fuente: www.xataka.com).

Todas estas situaciones, a excepción de la tercera, fueron extraídas de las encuestas realizadas. La tercera situación se identificó posteriormente en una reunión con los tutores de este trabajo de fin de grado.

5.2.1. Ontología

Una ontología es un formalismo empleado para representar conocimiento y compartir información en un área particular de conocimiento, incluyendo los términos clave, su interconexión semántica y ciertas reglas de inferencia. Una instancia de una ontología representa el conocimiento de la ontología con unos valores concretos. El propósito de una ontología es, por tanto, proporcionar una definición formal que dé una estructura semántica a los datos [38].

En este proyecto, la ontología se implementa para representar el conocimiento relativo a la conducción en un entorno urbano, identificando conceptos y sus propiedades y relaciones. Cada una de las instancias de esta ontología representará, por lo tanto, el estado de la situación de la conducción y su entorno en un instante de tiempo determinado. Esto ayuda al sistema a almacenar y estructurar la información para poder ser empleada en la identificación de situaciones de peligro y en la toma de decisiones.

Para definir una ontología, existe una gran variedad de lenguajes especializados. Entre los más conocidos se encuentran *RDF (Resource Description Framework)* [39], *RDF Schema* [40], *DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer)* [41], *OWL (Ontology Web Language)* [15] y *OWL-S* [42]. La ontología ya se encontraba implementada inicialmente por el grupo de investigación CAOS de la universidad Carlos III de Madrid, y el lenguaje empleado para la definición de la ontología fue *OWL*. Este lenguaje fue escogido sobre otros lenguajes para facilitar la representación de la información, ya que proporciona una noción representativa de la semántica para describir el contexto de la conducción.

OWL se basa en el lenguaje *XML*, y fue oficialmente estandarizado por *W3C (World Wide Web Consortium)* en febrero de 2004. Esta reciente estandarización es beneficiosa en términos de compatibilidad con proyectos existentes y futuros y de creación y soporte de herramientas software. Además, *OWL* proporciona una funcionalidad mayor que cualquier otro lenguaje de definición de ontologías.

La herramienta software compatible con este lenguaje utilizada fue *TopBraid Composer* [45]. Esta herramienta está basada en la plataforma *Eclipse* [46] y la *API Jena* [37], y se trata de un entorno de modelado visual para crear y administrar la ontología en lenguaje *OWL*.

Se muestra en la Figura 5.11 el esquema de la ontología heredada para este proyecto. Se observa en la Figura que esta ontología contiene los conceptos de coche, conductor, el contexto del coche, peatón y peatón cruzando (*Car*, *Driver*, *CarContext*, *PedCrossing* y *Pedestrian*). De cada uno de ellos se tienen los siguientes atributos con los siguientes valores:

- **Coche:**
 - **Distancia:** muy lejos, lejos, normal, cerca, muy cerca.
 - **Estado del vehículo:** en movimiento, parado.
- **Driver.**
 - **Área de visión:** frente, frente-derecha, frente-izquierda, izquierda, derecha, atrás-izquierda, atrás-derecha, retrovisor interior, retrovisor exterior izquierdo, retrovisor exterior derecho, velocímetro, radio/climatizador/abajo, techo(arriba).
 - **Atento:** sí, no.
 - **Dirección horizontal de los ojos:** centro, izquierda, derecha.
 - **Dirección vertical de los ojos:** centro, arriba, abajo.
- **CarContext.**
 - **Existencia de coche delante:** sí, no.
 - **Existencia de coche a la izquierda:** sí, no.
 - **Existencia de coche a la derecha:** sí, no.
 - **Existencia de peatón:** sí, no.
 - **Existencia de peatón cruzando:** sí, no.
- **PedCrossing.**
 - **Distancia al peatón:** muy lejos, lejos, normal, cerca, muy cerca.
- **Pedestrian.**
 - **Ángulo respecto del conductor:** frente, frente-izquierda, frente-derecha.
 - **Distancia:** muy lejos, lejos, normal, cerca, muy cerca.
 - **Trayectoria:** norte, sur, este, oeste, noreste, noroeste, sureste, suroeste.

El objetivo de este proyecto es la ampliación de dicha ontología para abarcar un rango más amplio de información y, con ello, poder abordar más situaciones, y más complejas, de peligro con el sistema de toma de decisiones basado en reglas. Se emplea la misma herramienta software, *TopBraid Composer ME*.

Surge el contratiempo de que la herramienta software mencionada ha dejado de ser capaz de tratar con archivos de ontología *OWL*, por lo que se traduce la ontología al lenguaje *RDF*, que si es compatible con las nuevas versiones de *TopBraid Composer ME*.

Se añade al contexto del coche la existencia de coches en más posiciones: detrás, detrás y a la izquierda, detrás y a la derecha, delante a la izquierda y delante a la derecha. Además, se incluye un nuevo concepto: el concepto del vehículo del conductor (*MyCar*). Este concepto presenta los siguientes atributos:

■ **MyCar**

- **Freno:** a fondo, sin pisar, medio.
- **Embrague:** a fondo, sin pisar, medio.
- **Marcha:** -1, 1, 2, 3, 4, 5, 6.
- **Aceleración lateral:** alta, media, baja, nula.
- **Aceleración longitudinal:** alta, media, baja, nula.
- **Revoluciones por minuto:** $[0, \text{maxRev}]$.
- **Velocidad:** alta, media, baja, nula. Se define velocidad media como valores aproximados a la máxima de la vía.
- **Angulo de giro del volante:** izquierda, medio-izquierda, centro, medio-derecha, derecha.
- **Acelerador:** a fondo, sin pisar, medio.

El esquema de la ontología completa queda, por lo tanto, como se muestra en la Figura 5.12.

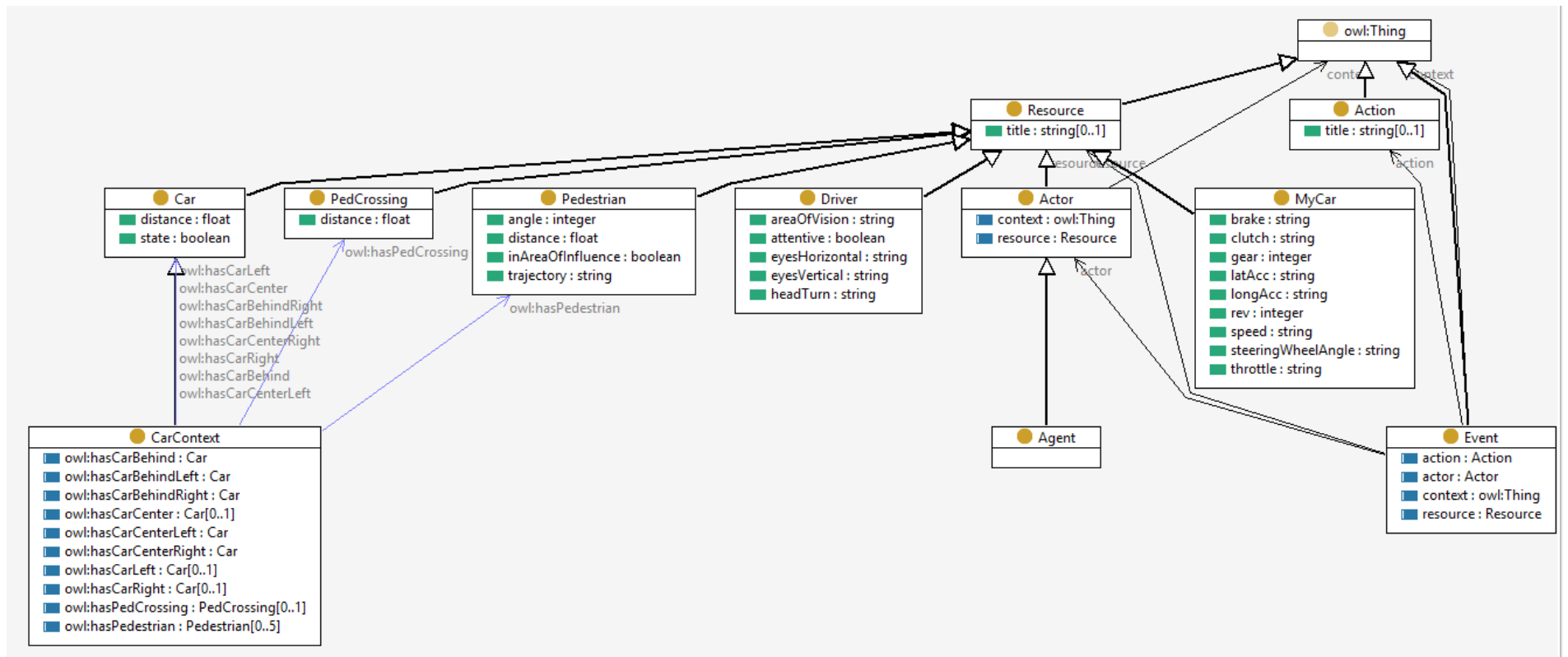


Figura 5.12: Ontología ampliada, sobre la del grupo de investigación CAOS, en este Trabajo de Fin de Grado.

Nótese que los valores de todos los atributos de los conceptos son nominales, lo que supone una limitación al sistema de toma de decisiones.

5.2.2. Gestión de consultas

La gestión de consultas tiene como objetivo reunir los datos necesarios acerca del estado de la vía, del vehículo y su entorno, y del conductor para poder ser utilizados finalmente en la parte de razonamiento. Para obtener dichos datos, es necesario realizar consultas a la ontología que organiza la información, descrita en el apartado anterior.

El lenguaje empleado para la realización de consultas es *SPARQL* [44] (acrónimo recursivo de *Protocol y RDF Query Language*). Se trata de un lenguaje de consultas semánticas que puede utilizarse tanto sobre ontologías *OWL* como *RDF*. Esto hace que este lenguaje sea adecuado para la implementación de la gestión de consultas.

Para realizar consultas utilizando este lenguaje, lo primero que se debe indicar son las ontologías de las propiedades que se van a utilizar. Esto se hace empleando la estructura *PREFIX* e indicando en ella las URLs correspondientes.

Una vez indicada la ontología o las ontologías que se vayan a utilizar, se procede a la definición de las demás consultas. Para ello se emplea la sentencia *SELECT*, la cual nos permite elegir los atributos que se mostrarán en el resultado, y la estructura *WHERE*, donde se indican las condiciones que devolverán el resultado. Cada condición se expresa con una tripla sujeto, predicado y objeto. Por ejemplo, una de las consultas implementadas se muestra en la Figura 5.13.

```
SELECT ?areaOfVision WHERE {?driver owl:areaOfVision ?areaOfVision.}
```

Figura 5.13: Ejemplo de consulta SPARQL.

En el ejemplo mostrado en la Figura 5.13, se desea como resultado el área de visión del conductor, el cual se indica como una variable (*?areaOfVision*). La restricción que se impone es que el atributo se encuentre en el concepto *driver*, es decir, en el conductor. Sin embargo, se indica una variable (*?driver*) para que se busque en todos los conceptos de la ontología (sujeto). Para producir un resultado, el atributo del concepto tiene que llamarse exactamente *areaOfVision* y estar dentro de la ontología *OWL* indicada en el prefijo (predicado). Por último, ese atributo tiene que tener como valor el de el área de visión del conductor, que se indica como una variable (*?areaOfVision*) para devolver cualquiera que sea el valor que tenga en ese momento (objeto). Esta variable tiene el mismo nombre que la variable que se desea como resultado para que se devuelva este valor.

Siguiendo este esquema de consultas, se realizan el resto de consultas necesarias para el razonamiento del sistema de toma de decisiones. Además de esta primera consulta acerca del área de visión del conductor, se tienen las siguientes consultas:

- Existencia de un coche delante, su distancia y su estado del vehículo.
- Existencia de un coche detrás, su distancia y su estado del vehículo.
- Existencia de un coche en diagonal hacia delante y a la derecha, su distancia y su estado del vehículo.
- Existencia de un coche en diagonal hacia delante y a la izquierda, su distancia y su estado del vehículo.
- Velocidad y ángulo de giro del volante del vehículo del conductor.
- De los peatones existentes, ángulo, distancia y trayectoria.

Una vez definidas todas las consultas, como ya se ha mencionado anteriormente, se ejecutan y se guardan los valores obtenidos para su utilización posterior en el razonamiento.

5.2.3. Razonamiento

Una vez definida y ampliada la ontología, y reunidos todos los datos necesarios, se definen las reglas que formarán la parte de razonamiento del sistema de toma de decisiones.

Para cada una de las cinco situaciones objeto de estudio descritas anteriormente, se definen una o varias reglas que avisen del peligro que está presente en cada una de ellas.

SITUACIÓN 1 - RIESGO DE COLISIÓN FRONTAL

Para esta situación, se comprueba:

- si existe un coche delante que esté cerca o muy cerca
- que el conductor no esté mirando al frente
- que la velocidad sea media o alta

Si estas condiciones se cumplen, el sistema lanzará la alarma: “Vehículo cercano o muy cercano no visualizado de frente”. Esta alarma está asociada, en el simulador, con la imagen mostrada en la Figura 5.14.



Figura 5.14: Imagen asociada a la alarma en la situación 1.

Esta imagen, y las demás que se indican, se mostrará a modo de advertencia en la interfaz del simulador.

SITUACIÓN 2 - RIESGO DE ATROPELLO

Para esta situación, se definen dos reglas. Una es para los peatones que cruzan la carretera de derecha a izquierda, y la otra es para los peatones que cruzan la carretera de izquierda a derecha.

En ambas reglas, se comprueba:

- que existan peatones visibles en la escena
- que los peatones estén al frente, o al frente y a la derecha, comprobando su ángulo
- que los peatones tengan trayectoria oeste (es decir, de derecha a izquierda), noroeste o suroeste, o, en la otra regla, este (es decir, de izquierda a derecha), noreste o sureste.
- que el conductor no esté mirando al frente
- que el peatón esté cerca o muy cerca del vehículo
- que la velocidad del vehículo sea media o alta

Si estas condiciones se cumplen, en cualquiera de las dos reglas, el sistema lanzará la alarma: “Peatón no visualizado cruza”. Esta alarma está asociada, en el simulador, con la imagen mostrada en la Figura 5.15.



Figura 5.15: Imagen asociada a la alarma en la situación 2.

SITUACIÓN 3 - RIESGO DE COLISIÓN TRASERA

Para esta situación, se comprueba:

- si existe un coche delante que esté cerca o muy cerca
- si existe un coche detrás que esté cerca o muy cerca
- que el coche de detrás esté en movimiento, comprobando su estado
- que el ángulo de giro del volante sea hacia la izquierda o medio-izquierda.

Si estas condiciones se cumplen, el sistema lanzará la alarma: “Atención al coche de atrás al adelantar”. Esta alarma está asociada, en el simulador, con la imagen mostrada en la Figura 5.16.

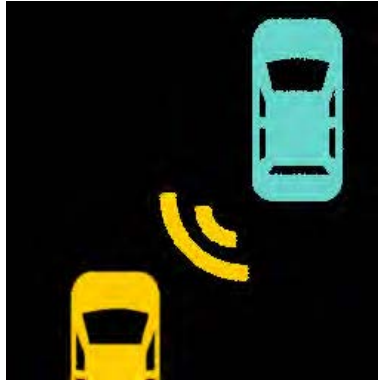


Figura 5.16: Imagen asociada a la alarma en la situación 3.

SITUACIÓN 4 - RIESGO DE COLISIÓN LATERAL

Para esta situación, se comprueba:

- si existe un coche delante o delante en diagonal a la derecha, parado, a una distancia normal, cercana o muy cercana
- que el conductor no esté mirando al frente

Si estas condiciones se cumplen, el sistema lanzará la alarma: “Coche detenido puede querer salir”. Esta alarma está asociada, en el simulador, con la imagen mostrada en la Figura 5.17.

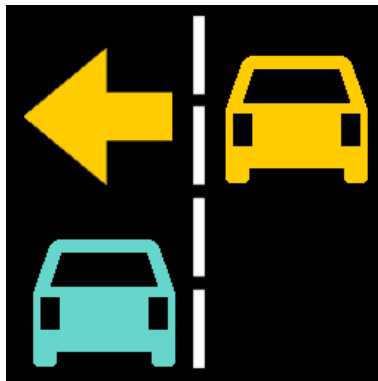


Figura 5.17: Imagen asociada a la alarma en la situación 4.

SITUACIÓN 5 - RIESGO DE ATROPELLO DE PEATÓN SIN VISUALIZAR

Para esta situación, nuevamente se definen dos reglas. Una es para los peatones no visibles al lado derecho de la carretera, que cruzan la carretera de derecha a izquierda, y la otra es para los peatones no visibles al lado izquierdo de la carretera, que cruzan la carretera de izquierda a derecha.

En la primera de las reglas, se comprueba:

- si existe un coche delante en diagonal a la derecha parado a una distancia normal, cercana o muy cercana
- la existencia de peatones
- que el peatón tenga una trayectoria hacia el oeste, noroeste o suroeste
- que el peatón esté al frente del vehículo
- que el conductor no esté mirando al frente

De forma análoga, en la segunda de las reglas, se comprueba:

- si existe un coche delante en diagonal a la izquierda parado a una distancia normal, cercana o muy cercana
- la existencia de peatones
- que el peatón tenga una trayectoria hacia el este, noreste o sureste
- que el peatón esté al frente del vehículo
- que el conductor no esté mirando al frente

Si estas condiciones se cumplen, en cualquiera de las dos reglas, el sistema lanzará la alarma: “Peatón no visible va a cruzar”. Esta alarma está asociada, en el simulador, con la imagen mostrada en la Figura 5.18.



Figura 5.18: Imagen asociada a la alarma en la situación 5.

Como el sistema sólo tiene una salida, en el caso de que dos reglas se disparen para un mismo instante de tiempo el sistema debería producir como salida aquella que indica un peligro mayor. Por lo tanto, se implementa también una jerarquía entre las diferentes alarmas disparadas por las reglas del sistema. Dicha jerarquía se expone en la tabla 5.1

Tabla 5.1: Jerarquía de alarmas (ordenadas de mayor a menor prioridad).

Situación	Alarma	Prioridad
Situación 2	Riesgo de atropello	1
Situación 1	Riesgo de colisión frontal	2
Situación 5	Riesgo de atropello de peatón sin visualizar	3
Situación 4	Riesgo de colisión lateral	4
Situación 3	Riesgo de colisión trasera	5

La situación más prioritaria se trata del riesgo de atropello, ya que está en juego la vida de una o más personas, y la vida es lo más prioritario. A continuación se tiene el riesgo de colisión frontal seguido del atropello de un peatón sin visualizar. Se sigue este orden ya que no se puede alcanzar a los peatones no visualizados si hay un vehículo delante a poca distancia. Después se tiene el riesgo de colisión lateral y, por último, el riesgo de colisión trasera, ya que solo se va a adelantar si no se produce ninguna otra situación de peligro.

Capítulo 6

Resultados y evaluación

En este capítulo de la memoria se procede a mostrar las pruebas realizadas sobre los sistemas implementados a fin de comprobar el correcto funcionamiento de los sistemas implementados y a fin de verificar que se cumple con los requisitos expuestos en el apartado 3 del documento, el cual corresponde al análisis del sistema.

Para cada una de las pruebas que se realicen, se empleará la tabla 6.1 para su especificación.

Tabla 6.1: Modelo para pruebas.

Identificador	P-XX
Descripción	
Objetivo	
Resultado esperado	
Resultado obtenido	
RF que cubre	

A continuación se explica qué significa cada una de las filas de la tabla:

- **Identificador.** Representa la manera de reconocer unívocamente a la prueba en cuestión para facilitar la trazabilidad. Los caracteres del identificador poseen el siguiente significado:
 - **P.** Prueba.
 - **XX.** El número entero positivo único que identifica cada prueba.
- **Descripción.** Corresponde a la explicación de la prueba que se va a realizar.
- **Objetivo.** Especifica qué funcionalidad se quiere probar.
- **Resultado esperado.** Especifica el resultado que se espera obtener de la ejecución de la prueba.
- **Resultado obtenido.** Expone el resultado obtenido, y si la prueba se ha realizado con éxito.

- **RF que cubre.** Indica los requisitos funcionales que se cubren con la realización correcta de la prueba en cuestión.

6.1. Gestión de rutas

Dentro de esta sección se describen las pruebas realizadas para el sistema de gestión de rutas (se dispone de un manual de usuario para la utilización de este módulo en el [Anexo I]). Para ello, se emplea la tabla modelo para pruebas (tabla 6.1) para cada una de las pruebas realizadas.

Las pruebas de la primera parte del proyecto, correspondientes al sistema de gestión de rutas, se exponen a continuación (Tablas de la 6.2 a la 6.9).

Tabla 6.2: P - 01.

Identificador	P - 01
Descripción	Se ejecuta el método de lectura de fichero de coordenadas <i>GPX</i> y se comprueban los valores almacenados en el programa.
Objetivo	Comprobar que la lectura del fichero se realiza de forma correcta.
Resultado esperado	Se leen los valores de latitud y longitud de cada uno de los puntos de la ruta y se almacenan para su uso posterior.
Resultado obtenido	Comprobados los valores almacenados y los valores del fichero, se determina que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 01.

Tabla 6.3: P - 02.

Identificador	P - 02
Descripción	Se ejecuta el método de transformación de coordenadas y se comprueban los valores almacenados en el programa.
Objetivo	Comprobar que la transformación de coordenadas del sistema <i>GPX</i> al sistema <i>UTM</i> se realiza correctamente.
Resultado esperado	Se realiza la transformación de forma correcta, y los valores almacenados en el programa son los correspondientes a los valores <i>GPX</i> transformados al sistema de coordenadas <i>UTM</i> .
Resultado obtenido	Se comprueba y se determina que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 02.

Tabla 6.4: P - 03.

Identificador	P - 03
Descripción	Se ejecuta el método que calcula la distancia entre cada par de puntos de coordenadas <i>UTM</i> .
Objetivo	Comprobar el correcto funcionamiento del método en cuestión.
Resultado esperado	Todas las distancias se han calculado, y los cálculos son correctos.
Resultado obtenido	Se comprueba que los cálculos y las distancias son correctos, por lo que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 03.

Tabla 6.5: P - 04.

Identificador	P - 04
Descripción	Se ejecuta el método que calcula el radio de las curvas en los puntos de coordenadas <i>UTM</i> .
Objetivo	Comprobar que los radios de las curvas del trazado se calculan correctamente.
Resultado esperado	Por cada curva existente en el trazado, el radio calculado por el sistema es correcto.
Resultado obtenido	Se comprueba que los radios calculados sean correctos y se concluye que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 04.

Tabla 6.6: P - 05.

Identificador	P - 05
Descripción	Se ejecuta el método que calcula el sentido de giro de las curvas del trazado. Desde este método se llama al método que calcula el ángulo entre dos vectores.
Objetivo	Comprobar el correcto funcionamiento de los métodos en cuestión.
Resultado esperado	Se determinan los sentidos de giro de las curvas de forma correcta, por lo que los ángulos entre los vectores se calculan correctamente también.
Resultado obtenido	Los cálculos son correctos, por lo que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 06, RF - 07.

Tabla 6.7: P - 06.

Identificador	P - 06
Descripción	Se ejecuta el método que combina los métodos de tratamiento de las curvas del trazado.
Objetivo	Comprobar que las curvas se calculan correctamente.
Resultado esperado	Los valores de los cálculos realizados sobre las curvas se han realizado y almacenado correctamente.
Resultado obtenido	Se comprueba que todos los cálculos y los valores almacenados son correctos, por lo que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 04, RF - 05, RF - 06, RF - 07, RF - 08.

Tabla 6.8: P - 07.

Identificador	P - 07
Descripción	Ejecutar el método de escritura del fichero con extensión .evt en lenguaje SDL del simulador, y abrir dicho fichero con el simulador de conducción.
Objetivo	Comprobar que el fichero se escribe correctamente y funciona en el simulador de conducción.
Resultado esperado	Se escribe el fichero en lenguaje SDL con extensión .evt, y el simulador es capaz de abrir y ejecutar dicho fichero.
Resultado obtenido	El fichero queda escrito, y el simulador es capaz de abrir y ejecutar el mismo, por lo que se determina que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 09.

Tabla 6.9: P - 08.

Identificador	P - 08
Descripción	Se ejecuta la versión de la aplicación que cuenta con una interfaz de usuario, se introduce la ruta del fichero de entrada y el fichero de salida, y se inicia la ejecución del programa.
Objetivo	Comprobar el correcto funcionamiento de la interfaz de usuario.
Resultado esperado	Se realiza todo el proceso de forma correcta, mostrando un aviso de ejecución sin problemas al terminar.
Resultado obtenido	Se comprueba que la ejecución se ha realizado correctamente, y el mensaje de confirmación aparece en la pantalla en una ventana emergente, por lo que se determina que el resultado de la prueba es satisfactorio.
RF que cubre	RF - 10, RF - 11, RF - 12, RF - 13.

6.2. Toma de decisiones

Dentro de esta sección se describen las pruebas realizadas para el sistema de toma de decisiones desarrollado en este trabajo de fin de grado. Para ello, se emplea la tabla modelo para pruebas (tabla 6.1) para cada una de las pruebas realizadas.

La realización de estas pruebas no es sencilla, puesto que no se realizan directamente sobre el código o sobre la interfaz de usuario. En estas pruebas, es necesario crear instancias de la ontología implementada en las que se den las situaciones que se quieran someter a prueba e introducirlas en el servicio web implementado por el grupo de investigación CAOS de la UC3M, el cuál dará la salida correspondiente según las reglas que forman parte del sistema de toma de decisiones presentado en el proyecto. Un ejemplo de prueba en el servicio web se muestra en la Figura 6.1.

SOAP request:	<pre> <soapenv:body> <urn:ADASTestAction soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" <frameData xsi:type="urn:complex"> <_time xsi:type="xsd:string">162404.223148</_time> <_driverData xsi:type="urn:complex"> <areaOfVision xsi:type="xsd:string">Izqda</areaOfVision> <eyesHorizontal xsi:type="xsd:string">Centro</eyesHorizontal> <eyesVertical xsi:type="xsd:string">Centro</eyesVertical> <headTurn xsi:type="xsd:string">-</headTurn> <attentive xsi:type="xsd:string">Si</attentive> </_driverData> <_myCarData xsi:type="urn:complex"> <speed xsi:type="xsd:string">Media</speed> <throttle xsi:type="xsd:string">Medio</throttle> <brake xsi:type="xsd:string">sinPisar</brake> <clutch xsi:type="xsd:string">sinPisar</clutch> <gear xsi:type="xsd:int">3</gear> <revolutions xsi:type="xsd:int">2000</revolutions> <steeringWheelAngle xsi:type="xsd:string">Centro</steeringWheelAngle> <longAcc xsi:type="xsd:string">Media</longAcc> <latAcc xsi:type="xsd:string">Media</latAcc> </_myCarData> <_carContextData xsi:type="urn:complex"> <_id xsi:type="xsd:string">1</_id> <_hasCarCenter xsi:type="urn:complex"> <id xsi:type="xsd:string">12</id> <distance xsi:type="xsd:string">MuyCerca</distance> <state xsi:type="xsd:string">EnMov</state> </_hasCarCenter> <_hasCarRight xsi:type="urn:complex"> <id xsi:type="xsd:string">12</id> </pre>
SOAP response:	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"><soapenv:Body><ADASTestActionResponse xmlns="urn:ADAS"><ADASTestActionReturn xmlns="">RESPONSE: ALARM2:(Vehiculo cercano o muy cercano no visualizado de frente)</ ADASTestActionReturn></ADASTestActionResponse></soapenv:Body></soapenv:Envelope> </pre>

Figura 6.1: Prueba de riesgo de colisión frontal en el servicio web.

En la parte de la Figura 6.1 señalada con un cuadro de color azul y la letra **A** se tiene un ejemplo de la instanciación de los valores de un concepto de la ontología. En este caso se trata del concepto del vehículo del conductor, *MyCar*. En la parte señalada con el cuadro verde y la letra **B** se tiene la respuesta del sistema de toma de decisiones basado en reglas según la situación planteada en la instancia de la ontología introducida como petición al sistema. En este caso, se ha disparado la alarma de riesgo de colisión frontal.

Además del servicio web, el agente se conecta con el simulador para comprobar el sistema de reglas en tiempo real. Mientras se está en el proceso de conducción, el simulador va creando instancias de la ontología con los valores correspondientes a cada momento y las va enviando como petición al sistema. De esta manera, las alarmas se disparan cada vez que se producen las situaciones de conducción de peligro que han sido objeto de estudio, como si de un coche real que tuviera el sistema implementado en él se tratara. Un ejemplo de alarma disparada mientras se está en proceso de conducción se muestra en la Figura 6.2.



Figura 6.2: Prueba de peatón cruza en el simulador de conducción.

Las pruebas de la segunda parte del proyecto, correspondientes al sistema de toma de decisiones basado en reglas, se expone a continuación (Tablas de la 6.10 a la 6.18). Las pruebas se han realizado a través del servicio web y del simulador. Sin embargo, se prefiere la realización de las pruebas específicas a través del servicio web debido a la comodidad que ofrece el poder diseñar los escenarios adecuados a cada una de las pruebas en lugar de provocar esas situaciones en la simulación.

Tabla 6.10: P - 01.

Identificador	P - 01
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, de riesgo de colisión frontal.
Objetivo	Comprobar que la situación de riesgo de colisión frontal funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: "ALARM2:(Vehiculo cercano o muy cercano no visualizado de frente)".
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-08.

Tabla 6.11: P - 02.

Identificador	P - 02
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, de riesgo de atropello.
Objetivo	Comprobar que la situación de riesgo de atropello funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM1:(Peaton no visualizado cruza)”.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-09.

Tabla 6.12: P - 03.

Identificador	P - 03
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, de riesgo de colisión trasera.
Objetivo	Comprobar que la situación de riesgo de colisión trasera funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM5:(Atencion al coche de atras al adelantar)”.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-10.

Tabla 6.13: P - 04.

Identificador	P - 04
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, de riesgo de colisión lateral.
Objetivo	Comprobar que la situación de riesgo de colisión lateral funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM4:(Coche detenido puede querer salir)”.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-11.

Tabla 6.14: P - 05.

Identificador	P - 05
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, de riesgo de atropello de peatón sin visualizar.
Objetivo	Comprobar que la situación de riesgo de atropello de peatón sin visualizar funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM3:(Peaton no visible va a cruzar)”.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-12.

Tabla 6.15: P - 06.

Identificador	P - 06
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, que contiene todos los riesgos descritos.
Objetivo	Comprobar que la jerarquía entre alarmas funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM1:(Peaton no visualizado cruza)”, correspondiente a la situación de mayor riesgo en este caso.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-13.

Tabla 6.16: P - 07.

Identificador	P - 07
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, que contiene los siguientes riesgos (en orden de mayor a menor prioridad): colisión frontal, atropello de peatón sin visualizar, colisión lateral, colisión trasera.
Objetivo	Comprobar que la jerarquía entre alarmas funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM2:(Vehiculo cercano o muy cercano no visualizado de frente)”, correspondiente a la situación de mayor riesgo en este caso.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-13.

Tabla 6.17: P - 08.

Identificador	P - 08
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, que contiene los siguientes riesgos (en orden de mayor a menor prioridad): atropello de peatón sin visualizar, colisión lateral, colisión trasera.
Objetivo	Comprobar que la jerarquía entre alarmas funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM3:(Peaton no visible va a cruzar)”, correspondiente a la situación de mayor riesgo en este caso.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-13.

Tabla 6.18: P - 09.

Identificador	P - 09
Descripción	Se realiza una petición al servicio web con un escenario, diseñado por el investigador, que contiene los siguientes riesgos (en orden de mayor a menor prioridad): colisión lateral, colisión trasera.
Objetivo	Comprobar que la jerarquía entre alarmas funciona correctamente.
Resultado esperado	El servicio web tiene como respuesta: “ALARM4:(Coche detenido puede querer salir)”, correspondiente a la situación de mayor riesgo en este caso.
Resultado obtenido	La respuesta del servicio web es la correcta, por lo que se determina que la prueba se ha pasado con éxito.
RF que cubre	RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07, RF-13.

Capítulo 7

Planificación del trabajo y entorno socioeconómico

En esta capítulo se describe la metodología software seguida para el desarrollo del proyecto, la planificación del trabajo, el presupuesto para la implementación de este proyecto completo en una empresa, estimando costes de personal y de equipo, tanto software como hardware, y el impacto socioeconómico, social y medioambiental esperado.

7.1. Metodología Software

En primer lugar, se define la metodología software a seguir. Para la realización de ambos subproyectos, tanto la gestión de rutas como la toma de decisiones, se sigue la metodología tradicional en cascada. Este modelo fue propuesto por Winston Royce en el año 1970, y posteriormente refinado por Boehm en 1981, Sommerville en 1985 y Sigwart y col. en 1990 [32].

En esta metodología software, el producto evoluciona a través de una secuencia de fases ordenadas de forma lineal, de manera que el inicio de la siguiente es el final de la anterior, permitiendo iteraciones al estado anterior. Después de cada etapa se realizan una o varias revisiones para comprobar si se puede pasar a la siguiente. Se emplea esta metodología por su facilidad de seguimiento y entendimiento, así como por su orientación a documentos.

Se expone a continuación el esquema seguido por el modelo en cascada (ver Figura 7.1):

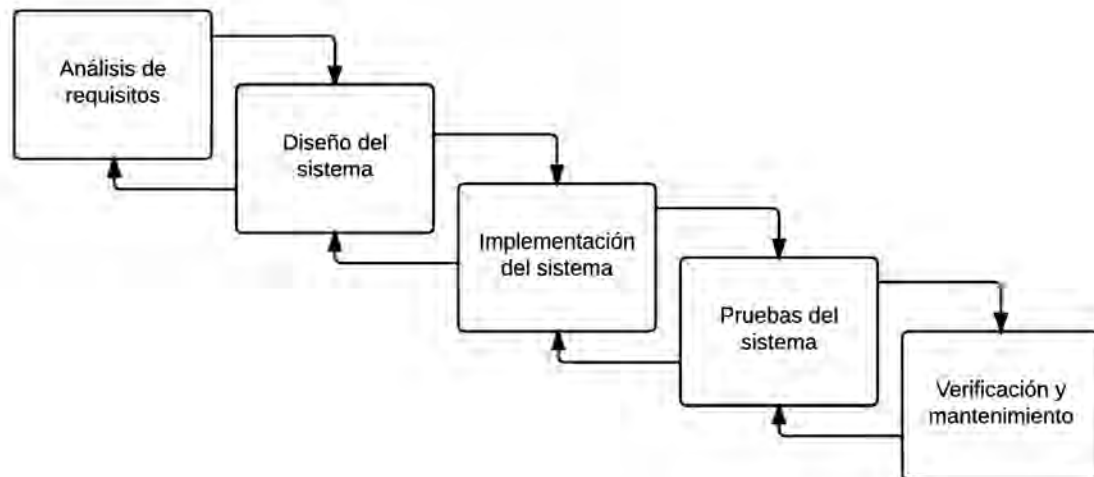


Figura 7.1: Planificación en cascada.

Como se observa en la Figura 7.1, las fases que componen esta metodología software son:

- **Análisis de requisitos.** En esta fase se estudian las necesidades del usuario final del software a fin de determinar qué objetivos debe alcanzar. Para ello, se programan reuniones, en este caso con los tutores del proyecto. El resultado de esta fase es la obtención de un documento de requisitos donde se detalle qué es lo que hay que hacer y cómo hay que hacerlo.
- **Investigación.** Esta fase no está incluida en el esquema del modelo clásico en cascada. Sin embargo, para la realización del presente proyecto es importante que el alumno realice un proceso de investigación acerca de cómo se debe abordar el problema y de qué herramientas van a ser usadas.
- **Diseño del sistema.** Después del análisis de los requisitos se llega al diseño del sistema. En esta fase, el sistema se divide en componentes o módulos especificando que debe hacer cada uno de ellos. El resultado de esta fase es, pues, la especificación de estos módulos en base a los requisitos obtenidos para su posterior implementación.
- **Implementación del sistema.** Una vez analizado el problema y diseñado la solución, ésta se codifica en código fuente.
- **Pruebas del sistema.** Después de implementar la solución diseñada, se prueba dicha implementación a fin de comprobar que cumple con los requisitos especificados en la primera fase y a fin de encontrar y solucionar los problemas que puedan aparecer.
- **Verificación y mantenimiento.** En esta fase el usuario final del software comprueba y verifica su buen funcionamiento. En caso de que existan problemas, hay un mantenimiento del software por parte del desarrollador.

- **Documentación.** Esta última fase tampoco está incluida en el esquema del modelo clásico en cascada. Sin embargo, es muy importante para este proyecto que, una vez finalizado, se documente todo en forma de memoria.

7.2. Planificación del trabajo

El presente proyecto tuvo su comienzo el día 28 de septiembre de 2015, y como fecha de finalización el día 14 de junio de 2016. De forma general, se muestra en la tabla 7.1 la duración total de cada subproyecto, tanto el sistema de gestión de rutas como el sistema de toma de decisiones, y la duración de la fase de documentación, indicando la fecha de comienzo y de finalización de cada trabajo.

Tabla 7.1: Tabla general de planificación del proyecto.

Trabajo	Duración (horas)	Fecha de comienzo	Fecha de finalización
Sistema de Gestión de Rutas	131	28/09/2015	15/01/2016
Sistema de Toma de Decisiones	181	18/01/2016	13/05/2016
Tarea de documentación	62	16/05/2016	14/06/2016

Así, el número total de horas de trabajo en el proyecto es 374. Se observa que los trabajos acaban en viernes, habiendo un pequeño descanso entre ellos del fin de semana correspondiente. Además, cabe destacar que durante el periodo de realización del Sistema de Gestión de Rutas hay un pequeño parón los días comprendidos entre el 8 y el 14 de enero, y en la tarea de documentación entre el 25 y el 31 de mayo, en ambos casos por motivos académicos.

Una vez indicada, de forma general, la duración de cada trabajo, se describe en detalle la planificación de cada subproyecto en las siguientes tablas. En cada una se indica: duración en horas, fecha de comienzo y final de cada etapa y, además, dentro de cada etapa, se especifica qué tareas hay y su duración en horas (debajo de cada etapa).

En primer lugar, se expone la tabla correspondiente a la planificación del sistema gestión de rutas (tabla 7.2).

Tabla 7.2: Planificación gestión de rutas.

Etapas	Duración (horas)	Fecha de comienzo	Fecha de finalización
ANÁLISIS DEL SISTEMA	15	28/09/2015	11/10/2015
Reuniones iniciales con tutor	4	28/09/2015	02/10/2015
Especificación de requisitos	6	29/09/2015	06/10/2015
Casos de uso	5	06/10/2015	11/10/2015
INVESTIGACIÓN	20	12/10/2015	25/10/2015
DISEÑO DEL SISTEMA	8	26/10/2015	04/11/2015
Definición de módulos	5	26/10/2015	02/11/2015
Interacción entre módulos	3	02/11/2015	04/11/2015
IMPLEMENTACIÓN DEL SISTEMA	61	05/11/2015	22/12/2015
Entrada-Salida	9	05/11/2015	12/11/2015
Transformación de coordenadas	11	13/11/2015	27/11/2015
Tratamiento de coordenadas <i>UTM</i>	34	28/11/2015	14/12/2015
Interfaz gráfica	7	15/12/2015	22/12/2015
PRUEBAS DEL SISTEMA	27	23/12/2015	15/01/2016
Especificación de las pruebas	4	23/12/2015	30/12/2015
Ejecución de las pruebas	23	02/01/2016	15/01/2016

A continuación, se expone la tabla correspondiente a la planificación del sistema de toma de decisiones (tabla 7.3).

Tabla 7.3: Planificación toma de decisiones.

Etapas	Duración (horas)	Fecha de comienzo	Fecha de finalización
ANÁLISIS DEL SISTEMA	35	18/01/2016	03/02/2016
Reuniones iniciales con tutor	8	18/01/2016	22/01/2016
Especificación de requisitos	15	20/01/2016	29/01/2016
Casos de uso	12	29/01/2016	03/02/2016
INVESTIGACIÓN	30	04/02/2016	23/02/2016
DISEÑO DEL SISTEMA	16	24/02/2016	11/03/2016
Definición de módulos	10	24/02/2016	08/03/2016
Interacción entre módulos	6	09/03/2016	11/03/2016
IMPLEMENTACIÓN DEL SISTEMA	69	14/03/2016	23/04/2016
Realización, distribución y evaluación de encuestas	12	14/03/2016	17/03/2016
Estudio y ampliación de la ontología	15	17/03/2016	25/03/2016
Gestión de consultas	21	25/03/2016	08/04/2016
Razonamiento	21	08/04/2016	23/04/2016
PRUEBAS DEL SISTEMA	31	25/04/2016	13/05/2016
Especificación de las pruebas	4	25/04/2016	28/04/2016
Implementación de las pruebas	5	29/04/2016	05/05/2016
Ejecución de las pruebas (simulador y web)	22	06/05/2016	13/05/2016

Se realiza un diagrama de Gantt, que muestra gráficamente toda la planificación del proyecto. Se expone en la Figura 7.2 y en la Figura 7.3. El primer diagrama muestra la planificación del sistema de gestión de rutas, y el segundo muestra la planificación del sistema de toma de decisiones y del proceso de documentación del proyecto total.

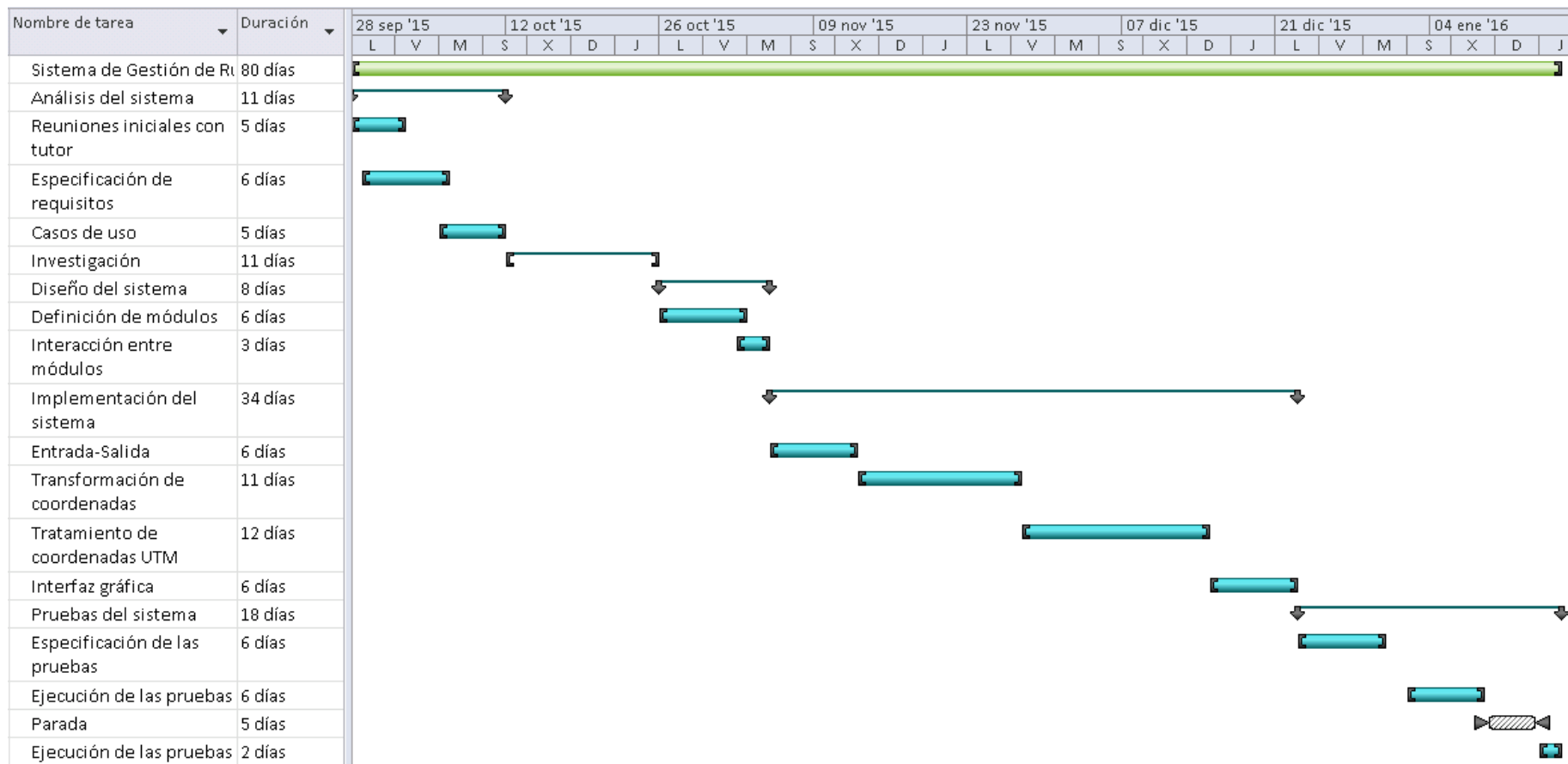


Figura 7.2: Planificación Gantt - Sistema de Gestión de Rutas.

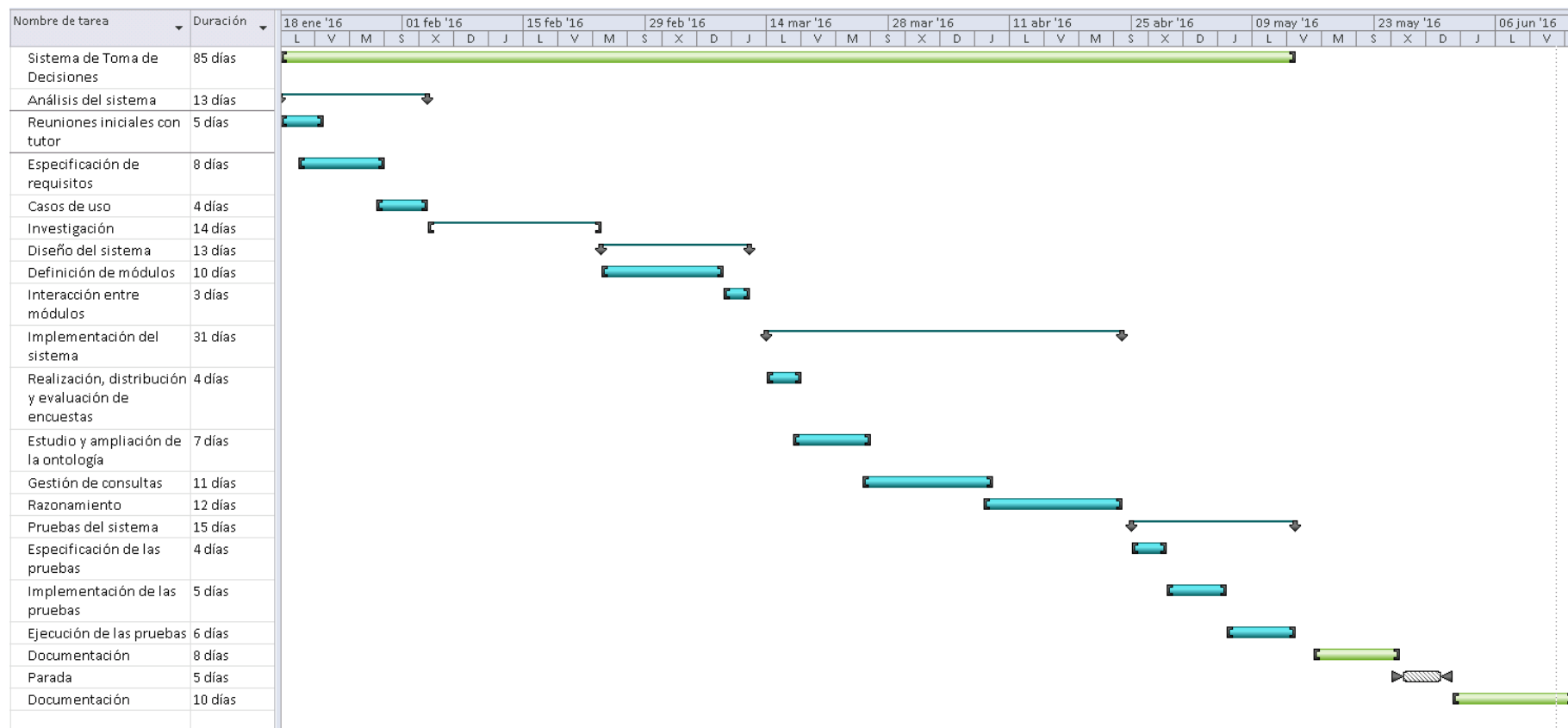


Figura 7.3: Planificación Gantt - Sistema de Toma de Decisiones y Documentación.

7.3. Presupuesto

En esta sección se presenta el presupuesto del proyecto que contiene a los dos sistemas descritos. Para ello, se analizarán los costes de personal, donde se identifican los roles de las personas que participan en el proyecto y se establecen sus sueldos, los costes de material, y los costes totales, teniendo en cuenta los costes indirectos.

7.3.1. Costes de personal

En primer lugar, se analizan los costes de personal. Para ello, se definen los roles presentes en el proyecto:

- **Jefe de proyecto.** Rol de la persona responsable de la dirección y la coordinación del proyecto.
- **Analista.** Rol de la persona responsable de la obtención de los requisitos del sistema.
- **Diseñador.** Rol de la persona responsable del diseño del sistema.
- **Programador.** Rol de la persona responsable de la codificación del sistema.
- **Técnico de pruebas.** Rol de la persona responsable de las pruebas del sistema.
- **Investigador.** Rol de la persona responsable de la documentación de todo el proyecto.

Una vez definidos los roles que participan en el proyecto, se procede a especificar cuáles serán sus salarios y el coste total de todos ellos. Esto se especifica en la tabla 7.4.

Tabla 7.4: Tabla de costes de personal.

Rol	Coste por hora (€)	Horas de trabajo	Total (€)
Jefe de proyecto	25,47	12	305,64
Analista	24,39	50	1.219,5
Diseñador	24,39	24	585,36
Programador	18,94	130	2.462,2
Técnico de pruebas	18,94	58	1.098,52
Investigador	16,04	112	1.796,48
Total	-	386	7.467,7

Los sueldos han sido establecidos a partir de la tabla salarial publicada en la resolución del 30 de enero de 2015, de la Dirección General de Empleo, por la que se registran y publican las tablas salariales correspondientes a los años 2013, 2014 y 2015 del Convenio colectivo de ámbito estatal para los centros de educación universitaria e investigación, BOE 11 de febrero de 2015.

7.3.2. Costes de material

A continuación, se analizan los costes de los materiales empleados en el desarrollo del proyecto. Para ello, es necesario definir cuáles son los productos utilizados, su precio, el periodo de amortización de cada uno de los productos, el tiempo de uso en el proyecto y, finalmente, el coste de ese material para el presente proyecto. Dicho coste se calcula de la siguiente forma:

$$coste = \frac{precio}{periodo\ amortización} \cdot uso \quad (7.1)$$

En la tabla 7.5 se muestran los costes de material (sólo se incluyen aquellos productos utilizados que supongan un coste de dinero):

Tabla 7.5: Tabla de costes de material.

Producto	Precio (€)	Periodo amortización (meses)	Tiempo uso (meses)	Coste en proyecto (€)
PC simulador	900,08	48	8	150,01
PC desarrollo	719,00	48	8	119,83
STI SIM M100	7.500,00	48	8	1.250,00
KINECT 2.0	149,99	48	8	24,99
Logitech G27 (HW conducción)	290,00	48	8	48,33
Licencia Office 2013 Professional	539,00	60	8	71,87
Licencia TopBraid Composer ME	3.060,69	60	4	204,04
Licencia Microsoft Project Prof	919,00	60	1	15,32
Total	-	-	-	1.884,39

7.3.3. Costes totales

Una vez estudiados los costes de personal y los costes de material, que conforman los costes directos del proyecto, es necesario añadir los costes indirectos. Estos serán el 10 % del total, y representarán costes tales como la electricidad, el acceso a internet, el teléfono, etc.

Además, también se aplica un porcentaje de 15 % del coste total como margen de riesgo para cubrir posibles imprevistos, y otro porcentaje del 15 % sobre el coste total como beneficios del proyecto. Además, por último, es necesario aplicar el porcentaje de IVA sobre el total, que es del 21 %.

El resumen de coste total se expone en la tabla 7.6.

Tabla 7.6: Tabla de costes totales.

Concepto	Cantidad (€)
Total costes directos	9.352,09
Costes indirectos (10 %)	935,21
Margen de riesgo (15 %)	1.402,81
Beneficios (15 %)	1.402,81
Base imponible	13.092,92
IVA (21 %)	2.749,51
Total coste del proyecto	15.842,43

7.4. Impacto socioeconómico esperado

En esta sección se procede a analizar el impacto socioeconómico de la aplicación del resultado del proyecto. Ambos subproyectos, la gestión de rutas y la toma de decisiones, tienen como objetivo formar parte de un Sistema Avanzado de Asistencia a la Conducción. Por lo tanto, la aplicación del resultado del proyecto sería ayudar en las tareas de circulación de los conductores estando incorporado en el vehículo, y el sector de aplicación sería el sector de la automoción.

El impacto que se espera que este proyecto genere es aportar una característica nueva a los vehículos que mejore, sobre todo, la seguridad de la conducción. Esto provocaría un aumento en el precio de los vehículos, puesto que la tecnología incorporada sería mayor y el coste del desarrollo y de la producción aumentaría. Sin embargo, si se tiene en cuenta el hecho de que esta tecnología reduciría el número de accidentes de tráfico, puede suponer un ahorro de dinero a la larga tanto para los conductores como para el estado, puesto que disminuiría el número de coches siniestrados, el número de reparaciones de vehículos y el número de desperfectos producidos por los accidentes. Además, otro beneficiado serían las compañías aseguradoras, que reducirían el número de indemnizaciones a pagar, aumentando así su margen de beneficio.

Sin embargo, dentro del sector automovilístico, la aplicación del resultado de este proyecto puede afectar de manera negativa a los talleres de reparaciones, puesto que al evitarse cierto porcentaje de accidentes tendrían menos reparaciones que realizar, lo que reduciría su margen de beneficio, y a concesionarios y casas de venta de vehículos, puesto que descendería el número de vehículos que quedan inutilizados por accidente, haciendo que surja la necesidad de comprar un coche nuevo.

7.5. Impacto social y medioambiental

Como ya se ha mencionado en el apartado anterior, una de las implicaciones de la aplicación del resultado del proyecto es la disminución del número de accidentes de tráfico. Este hecho no solo tiene un impacto positivo a nivel económico, como se ha visto anteriormente, sino que también lo tiene a nivel social y medioambiental.

En los accidentes de tráfico, más importante que el dinero son las vidas que hay en juego. Al disminuir éstos, el número de vidas que se pierden en las carreteras sería menor. Por lo tanto, se tendría un impacto social muy positivo.

En cuanto al impacto en el medio ambiente, se tiene que éste también es positivo. Esto es debido a que el hecho de que el número de accidentes de tráfico se vea reducido hace que se reduzca la pérdida de los materiales que conforman los vehículos y las vías donde se producen, así como todos los objetos que tengan riesgo de romperse. Por lo tanto, se tiene que la vida útil de los vehículos se vería aumentada y no se malgastan materiales en reparaciones que pueden ser evitadas.

Capítulo 8

Conclusiones y trabajos futuros

En este último capítulo de la memoria del trabajo de fin de grado, se exponen las conclusiones extraídas de la realización del mismo, tanto técnicas como personales, y se presentan las líneas futuras de trabajo que se pueden seguir partiendo de este proyecto.

8.1. Conclusiones técnicas

Tras la realización del presente proyecto, se concluye que los objetivos propuestos al comienzo del mismo han sido cumplidos. Por un lado se ha conseguido desarrollar el módulo de gestión de rutas capaz de leer e interpretar ficheros de rutas expresadas en coordenadas *GPX*, transformar dichas coordenadas al sistema de coordenadas *UTM*, identificar el trazado a partir de esas coordenadas *UTM*, generar un fichero de salida escrito en lenguaje *SDL* del simulador con el escenario dado por la ruta, y que además cuenta con una interfaz gráfica que permite configurar los parámetros de la aplicación.

Por otro lado, se ha estudiado el sistema de agentes ya implementado, se ha ampliado la ontología existente para disponer de una información más completa por cada toma de muestra, y se ha conseguido desarrollar el sistema de toma de decisiones basado en reglas, que es capaz de realizar consultas sobre la ontología ampliada para reunir información y emplearla para implementar el sistema de reglas que lance alarmas.

De forma general, se destaca que el alumno ha sido capaz de aplicar sus conocimientos de una forma profesional a un problema relacionado con el ámbito de la ingeniería informática, reuniendo e interpretando datos relevantes dentro del área de estudio para emitir juicios que incluyen una reflexión sobre un tema relevante de índole científico, como es la seguridad vial y los sistemas avanzados de asistencia a la conducción.

8.2. Conclusiones personales y conocimientos adquiridos

Con respecto a las conclusiones personales extraídas, es satisfactorio y motivo de orgullo el haber realizado este proyecto que cierra mis estudios en el grado en Ingeniería Informática, puesto que se han cumplido todos los objetivos propuestos al principio.

Al realizar un proyecto software de tales dimensiones, se ha aprendido acerca de la dificultad y de la importancia de planificar el proyecto definiendo objetivos y tiempos de realización, e ir sacándolo adelante ajustándose a lo especificado. Además, la realización del proyecto ha supuesto un acercamiento al trabajo de investigación que es llevado a cabo por grupos de investigación como es el grupo con el que se ha trabajado, CAOS, y ha permitido al alumno desarrollar habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

También se realiza un aprendizaje en las propias tareas realizadas. Un ejemplo de esto es el aprendizaje del uso de la herramienta *TopBraid Composer ME* para el manejo de la ontología, los lenguajes *RDF*, *OWL*, *SPARQL* para la definición y consultas sobre esa ontología, o el lenguaje *SDL* para la definición de escenarios del simulador de conducción utilizado.

8.3. Trabajos futuros

Este proyecto representa una primera aproximación a un sistema de toma de decisiones que forma parte de un sistema avanzado de asistencia a la conducción, por lo que el objetivo final del mismo es ir incorporado en un vehículo real. Como trabajo futuro principal destaca la posibilidad de completar el sistema aumentando el número de situaciones de peligro a considerar y advertir por el mismo, así como completar las situaciones ya existentes. Además, también se podrían diferenciar escenarios dentro de la misma situación, donde la gravedad aumenta a medida que la distancia disminuye.

El hecho de hacer que el sistema tenga en cuenta más situaciones lleva ligada la necesidad de ampliar la ontología para que ésta recoja más datos de la tarea de la conducción, tanto del conductor, como del vehículo y su entorno, y el sistema de reglas.

Una de las limitaciones del sistema es que los valores de los atributos de los conceptos son categóricos. Una mejora del sistema sería hacer que los datos que se obtienen del proceso de conducción sean numéricos, lo que haría que las reglas fueran mucho más precisas.

Una limitación más es que sólo se puede tener una salida, lo que implica la necesidad de establecer una jerarquía de alarmas, que corresponden a la salida del sistema. Si se amplía el número de situaciones a tener en cuenta, también se produce un aumento de la complejidad de dicha jerarquía. Es por eso que una de las líneas de trabajos futuros es la implementación del mismo sistema mediante el empleo de la lógica difusa, la cual permite considerar todas las reglas a la vez y prescindir así de la jerarquía de alarmas.

En cuanto al módulo de gestión de rutas implementado en este proyecto, éste tiene la limitación de que el sistema solo es capaz de leer los ficheros de rutas con un formato determinado. Como trabajo futuro se podría dar la capacidad al sistema de leer las rutas en más formatos y sistemas de coordenadas.

Otra limitación del sistema es que la salida del mismo sólo representa en el simulador la ruta a recorrer, pero no el entorno de la carretera. Es decir, se muestra el entorno de conducción que hay por defecto. Como trabajo futuro, se podría generar aleatoriamente un entorno con decoración alrededor y un estilo de vía acorde, tanto en su número de carriles y el sentido de los mismos como en su pavimentación.

Por último, también sería útil e interesante hacer que el sistema de gestión de rutas mostrara gráficamente, a partir de las coordenadas recibidas, el trazado de la ruta que se va a representar en lenguaje *SDL*.

Bibliografía

- [1] OMS, *Informe sobre la situación actual de la seguridad vial*, 2015
- [2] AUTOCONNECTEDCAR.COM, *ADAS Definition*, 2012. URL: <http://tinyurl.com/j8qe6ce> (Último acceso: 30/03/2016).
- [3] GERMÁN GUTIERREZ, JOSÉ ANTONIO IGLESIAS, FRANCISCO JAVIER ORDOÑEZ, AGAPITO LEDEZMA, ARACELI SANCHIS, *Agent-Based Framework for Advanced Driver Assistance Systems in Urban Environments*, 2014.
- [4] SYSTEMS TECHNOLOGY, INC., *STISIM Drive: Car Driving Simulator & Simulation Software*. URL: <http://www.stisimdrive.com/products/simulation-systems/m100-series> (Último acceso: 29/03/2016).
- [5] REDACCIÓN NOTICIAS.COCHES.COM, *¿Qué es el control de velocidad adaptativo?*, 2009. URL: <http://noticias.coches.com/noticias-motor/que-es-el-control-de-velocidad-adaptativo/3833> (Último acceso 17/05/2016).
- [6] JAVIER COSTAS, *Cómo ha evolucionado la seguridad de los coches en 40 años (II)*, 2008. URL: <http://www.circulaseguro.com/como-ha-evolucionado-la-seguridad-de-los-coches-en-40-anos-ii/> (Último acceso: 17/05/2016).
- [7] COLABORACIÓN RACC - ADAC, *Sistemas de reconocimiento de señales de tráfico en turismos*, 2011. URL: http://imagenes.w3.racc.es/uploads/file/22207_Sistema_Reconocimiento_Seniales.pdf (Último acceso: 17/05/2016).
- [8] IRENE MENDOZA, *¿Qué es el sistema de mantenimiento de carril?*, 2013. URL: <http://www.motor.mapfre.es/consejos-practicos/seguridad-vial/4183/que-es-el-sistema-de-mantenimiento-de-carril> (Último acceso: 17/05/2016).
- [9] EURONCAP, *Frenado de emergencia autónomo*, 2016. URL: <http://www.euroncap.com/es/seguridad-en-los-veh%C3%ADculos/descripci%C3%B3n-de-los-premios/frenado-de-emergencia-aut%C3%B3nomo/> (Último acceso: 17/05/2016).
- [10] DANI MEGANEBOY, *Sistema de Aparcamiento Asistido*, 2014. URL: <http://www.aficionadosalamecanica.com/sistema-aparcamiento-asistido.htm> (Último acceso: 17/05/2016).

- [11] IBÁÑEZ, *Asistente de mantenimiento en carril: ¿comodidad o seguridad?*, 2013. URL: <http://www.circulaseguro.com/asistente-de-mantenimiento-en-carril-comodidad-o-seguridad/> (Último acceso: 17/05/2016).
- [12] DOUG NEWCOMB, *Active Auto Safety gets in your face*, 2012. URL: <http://www.wired.com/> (Último acceso: 20/06/2016).
- [13] TIM STEVENS, *Toyota's New Eyelid-Monitoring System Wakes Up Sleepy Drivers*, 2008. URL: <http://www.switched.com/> (Último acceso: 17/05/2016).
- [14] K. TORKKOLA, M. GARDNER, C. SCHREINER, K. ZHANG, B. LEIVIAN, H. ZHANG, J. SUMMERS, *Understanding driving activity using ensemble methods. En Danil Prokhorov (Ed.) Computational Intelligence in Automotive Applications Studies in Computational Intelligence Volume 132 (pp 39-58). Springer Berlin Heidelberg.*, 2008.
- [15] W3C SEMANTIC WEB, *Web Ontology Language (OWL)*, 2016. URL: <https://www.w3.org/OWL/> (Último acceso 16/06/2016).
- [16] SIMUMAK SIMULATION & TRAINING SOLUTIONS, *Escuela de conducción de coche Simescar*, 2016. URL: <http://simumak.com/es/simescar> (Último acceso: 17/05/2016).
- [17] THE NATIONAL ADVANCED DRIVING SIMULATOR, *NADS Simulators NADS-1*, 2014. URL: https://www.nads-sc.uiowa.edu/sim_nads1.php (Último acceso: 17/05/2016).
- [18] DRIVESIM SIMULATION S.A., *DriveSim*, 2016. URL: <http://drivesimsimulator.com/> (Último acceso: 17/05/2016).
- [19] OCULUS VR, LLC, *Oculus Rift - Next-generation virtual reality*, 2016. URL: <https://www.oculus.com/en-us/rift/> (Último acceso: 15/06/2016).
- [20] CITY CAR DRIVING - CAR DRIVING SIMULATOR, PC GAME, *City Car Driving*, 2016. URL: <http://citycardriving.com/> (Último acceso: 17/05/2016).
- [21] AUTOESCUELA PRADO, *Simulador*, 2015. URL: <http://www.autoescuelaprado.es/simulador.html> (Último acceso: 17/05/2016).
- [22] HAMMACHER SCHLEMMER, *The Most Realistic Racing Simulator*, 2016. URL: <http://www.hammacher.com/Product/12677?promo=search&cmp=cel&trigger=ac&query=the%20most> (Último acceso: 17/05/2016).
- [23] TURN 10 STUDIOS, *Forza Motorsport*, 2016. URL: <http://forzamotorsport.net/en-us/> (Último acceso: 17/05/2016).
- [24] SLIGHTLY MAD STUDIOS, *Project Cars Game*, 2015. URL: <http://www.projectcarsgame.com/> (Último acceso: 17/05/2016).
- [25] SONY INTERACTIVE ENTERTAINMENT LLC, *PlayStation VR – Virtual Reality Headset for PS4*, 2016. URL: <https://www.playstation.com/en-us/explore/playstation-vr/> (Último acceso: 15/06/2016).

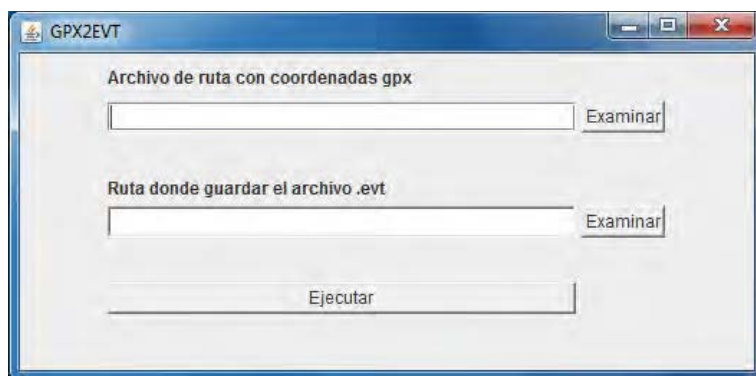
- [26] DGT, *Instrucción 15/V-113 - Autorización de pruebas o ensayos de investigación realizados con vehículos de conducción automatizada en vías abiertas al tráfico en general*, 2016.
- [27] BOE, *Real Decreto 2822/1998, de 23 de diciembre, por el que se aprueba el Reglamento General de Vehículos*, 1999.
- [28] BOE, *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.*, 2000.
- [29] OSRM, *Open Source Routing Machine*. URL: <http://map.project-osrm.org> (Último acceso: 29/03/2016).
- [30] ORS, *Open Route Service*. URL: <http://openrouteservice.org/> (Último acceso: 29/03/2016).
- [31] SVERRIR SIGMUNDARSON, *Google Maps to GPX converter*. URL: <http://labs.coruscantconsulting.co.uk/garmin/gpxgmap/convert.php> (Último acceso: 29/03/2016).
- [32] CATALDI, Z., LAGE, F., PESSACQ, R. Y GARCÍA MARTÍNEZ, R., *Ingeniería de software educativo*.
- [33] STISIM DRIVE, *Scenario Definition Language Events Manual*.
- [34] SANTOS DAMIAN, *Aprende a convertir coordenadas geograficas en UTM y UTM en geograficas*, 2004. URL: http://www.academia.edu/9291952/Aprende_a_convertir_coordenadas_geograficas_en_UTM_y_UTM_en_geograficas (Último acceso: 07/06/2016).
- [35] JADE SITE, *JAVA Agent DEvelopment Framework*, 2016. URL: <http://jade.tilab.com/> (Último acceso: 16/06/2016).
- [36] FIPA, *Welcome to FIPA!*, 2016. URL: <http://www.fipa.org/> (Último acceso: 14/06/2016).
- [37] THE APACHE SOFTWARE FOUNDATION, *Apache Jena*, 2016. URL: <https://jena.apache.org/> (Último acceso: 14/06/2016).
- [38] PÉREZ SOLTERO, BARCELO VALENZUELA, SÁNCHEZ SCHMIDTZ, NAVARRO HERNÁNDEZ, *Diseño de una Ontología para la Reutilización del Conocimiento en los Procesos de Auditoría del Conocimiento*, 2008.
- [39] W3C SEMANTIC WEB, *Resource Description Framework (RDF)*, 2016. URL: <https://www.w3.org/RDF/> (Último acceso: 16/06/2016).
- [40] W3C SEMANTIC WEB, *RDF Schema 1.1*, 2016. URL: <https://www.w3.org/TR/rdf-schema/> (Último acceso: 16/06/2016).
- [41] W3C SEMANTIC WEB, *DAML+OIL (March 2001) Reference Description*, 2001. URL: <https://www.w3.org/TR/daml+oil-reference> (Último acceso: 16/06/2016).
- [42] W3C SEMANTIC WEB, *OWL-S: Semantic Markup for Web Services*, 2004. URL: <https://www.w3.org/Submission/OWL-S/> (Último acceso: 16/06/2016).
- [43] STISIM DRIVE, *STISIM Drive - Introduction*. URL: http://web.mit.edu/16.400/www/auto_sim/Help/Introduction.htm (Último acceso: 16/06/2016).

- [44] W3C SEMANTIC WEB, *SPARQL Query Language for RDF*, 2007. URL: <https://www.w3.org/TR/rdf-sparql-query/> (Último acceso: 16/06/2016).
- [45] W3C SEMANTIC WEB, *TopBraid*, 2011. URL: <https://www.w3.org/2001/sw/wiki/TopBraid> (Último acceso: 16/06/2016).
- [46] THE ECLIPSE FOUNDATION OPEN SOURCE COMMUNITY WEBSITE., *Eclipse*, 2016. URL: <https://eclipse.org/> (Último acceso: 16/06/2016).

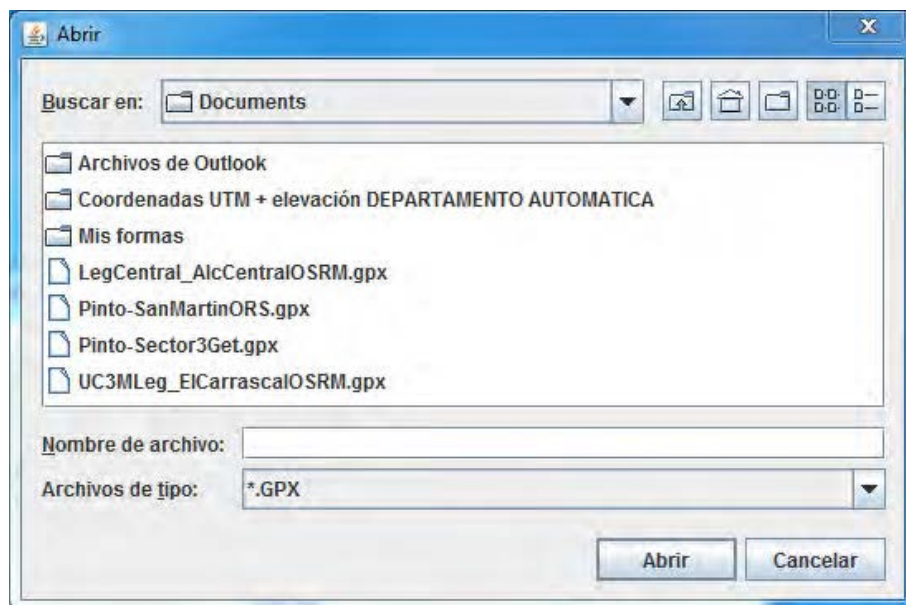
Anexos

Anexo I: Manual de usuario - Sistema de Gestión de Rutas (uso de la aplicación)

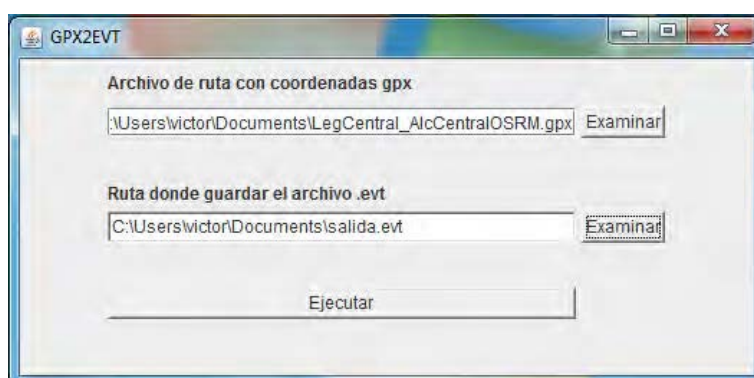
A continuación se presenta el manual de usuario de la aplicación implementada de gestión de rutas. El primer paso a seguir es la ejecución de la aplicación en formato *jar*. Una vez hecho esto, se muestra la interfaz de la aplicación, como se muestra en la siguiente imagen.



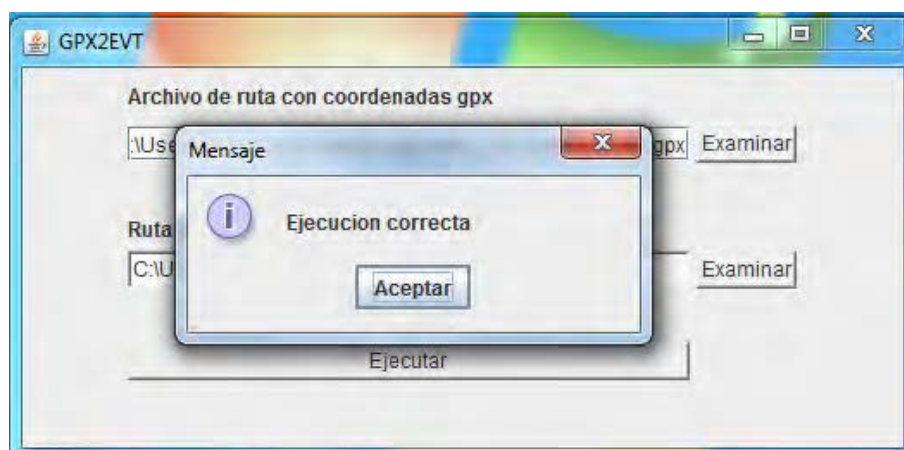
Esta interfaz permite introducir la ruta del archivo de entrada con la ruta en coordenadas *GPX* de dos maneras diferentes: introduciéndola de manera manual o haciendo click en el botón “Examinar”. Realizando la segunda de las maneras expuestas, se muestra la siguiente ventana para poder navegar por el sistema de ficheros:



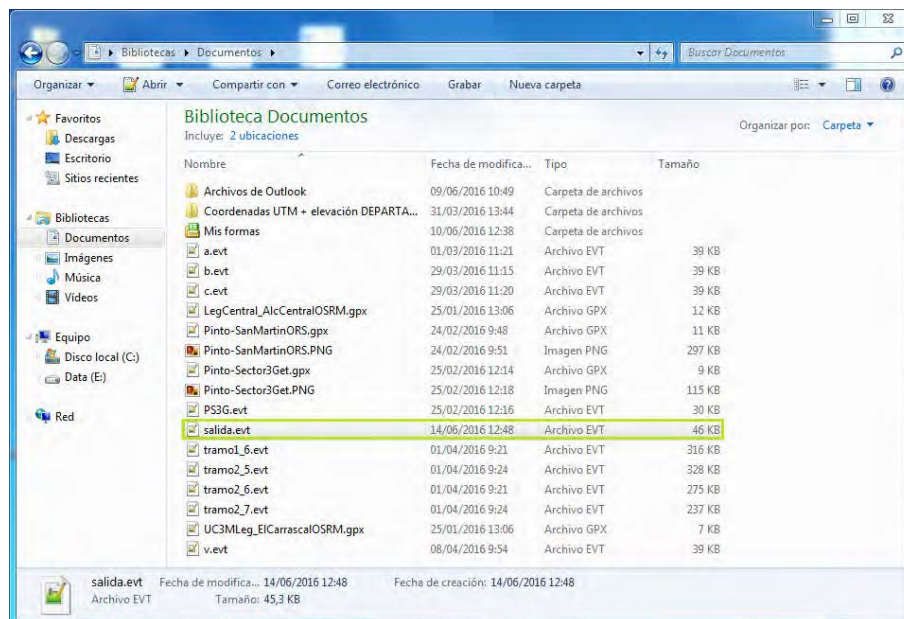
Una vez seleccionada la ruta del fichero que contiene las coordenadas de entrada, se procede de la misma forma para introducir la ruta donde se encuentra el fichero de salida o, si no existe, donde se creará dicho fichero. Se muestra a continuación la interfaz con las dos rutas introducidas:



Ya introducidos los parámetros, se hace click en el botón “Ejecutar”, y, si no hay ningún error durante el proceso, se muestra una notificación emergente indicando que la ejecución ha sido correcta.



Por último, se observa que el archivo de salida se ha creado correctamente en la ruta indicada.

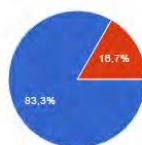


Anexo II: Encuestas acerca de situaciones de peligro en conducción por ciudad

Situación 1

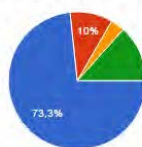
Estás distraído en una carretera con mucho tráfico, y la distancia al coche de delante se recorta llegando a ser muy pequeña.

¿Te ha pasado alguna vez?



Sí	25	83.3%
No	5	16.7%

¿Qué tipo de peligro crees que representa esta situación?



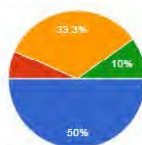
Peligro alto	22	73.3%
Peligro medio	3	10%
Peligro bajo	1	3.3%
Depende de la distancia	4	13.3%

¿Te gustaría que un Sistema Avanzado de Asistencia al Conductor te avisara de esta situación en caso de no estar atento a la carretera?



Sí	29	96.7%
No	1	3.3%

En caso positivo ¿Cómo?



Sonido de intensidad dependiente del nivel de peligro	15	50%
Luz en el cuadro del coche	2	6.7%
Sonido y luz	10	33.3%
Otro	3	10%

Situación 2

Estás conduciendo por la ciudad cuando un peatón cruza por donde no debe.

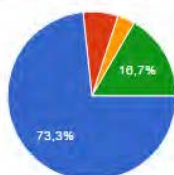
¿Te ha pasado alguna vez?

Si	30	100%
No	0	0%



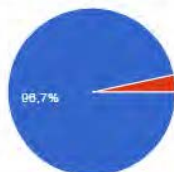
¿Qué tipo de peligro crees que representa esta situación?

Peligro alto	22	73.3%
Peligro medio	2	6.7%
Peligro bajo	1	3.3%
Depende de la distancia	5	16.7%



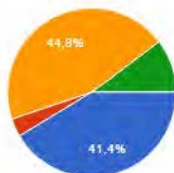
¿Te gustaría que un Sistema Avanzado de Asistencia al Conductor te avisara de esta situación en caso de no estar atento a la carretera?

Si	29	96.7%
No	1	3.3%



En caso positivo ¿Cómo?

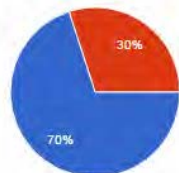
Sonido de intensidad dependiente del nivel de peligro	12	41.4%
Luz en el cuadro del coche	1	3.4%
Sonido y luz	13	44.8%
Otro	3	10.3%



Situación 3

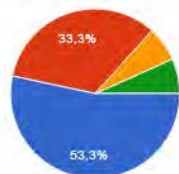
Vas a adelantar a un vehículo, viene otro vehículo de frente y tienes que volver al carril rápidamente.

¿Te ha pasado alguna vez?



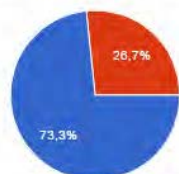
Sí	21	70%
No	9	30%

¿Qué tipo de peligro crees que representa esta situación?



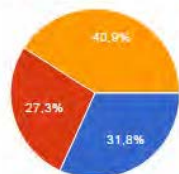
Peligro alto	16	53.3%
Peligro medio	10	33.3%
Peligro bajo	2	6.7%
Depende de la distancia	2	6.7%

¿Te gustaría que un Sistema Avanzado de Asistencia al Conductor te avisara de esta situación en caso de no estar atento a la carretera?



Sí	22	73.3%
No	8	26.7%

En caso positivo ¿Cómo?

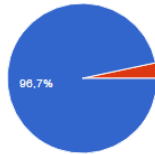


Sonido de intensidad dependiente del nivel de peligro	7	31.8%
Luz en el cuadro del coche	6	27.3%
Sonido y luz	9	40.9%
Otro	0	0%

Situación 4

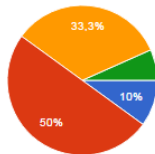
Hay un coche aparcado que sale sin señalizarlo con los intermitentes, pudiendo provocar una situación de peligro.

¿Te ha pasado alguna vez?



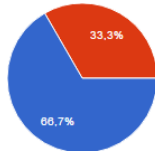
Si	29	96.7%
No	1	3.3%

¿Qué tipo de peligro crees que representa esta situación?



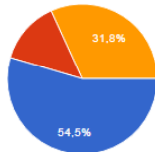
Peligro alto	3	10%
Peligro medio	15	50%
Peligro bajo	10	33.3%
Depende de la distancia	2	6.7%

¿Te gustaría que un Sistema Avanzado de Asistencia al Conductor te avisara de esta situación en caso de no estar atento a la carretera?



Si	20	66.7%
No	10	33.3%

En caso positivo ¿Cómo?

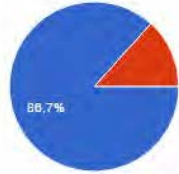


Sonido de intensidad dependiente del nivel de peligro	12	54.5%
Luz en el cuadro del coche	3	13.6%
Sonido y luz	7	31.8%
Otro	0	0%

Situación 5

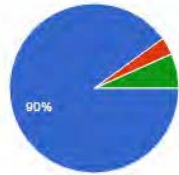
Estás conduciendo por la ciudad cuando un peatón sale de detrás de un coche aparcado.

¿Te ha pasado alguna vez?



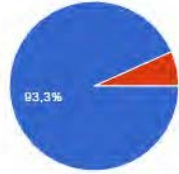
Sí	26	86.7%
No	4	13.3%

¿Qué tipo de peligro crees que representa esta situación?



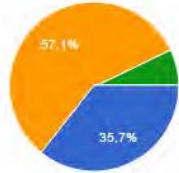
Peligro alto	27	90%
Peligro medio	1	3.3%
Peligro bajo	0	0%
Depende de la distancia	2	6.7%

¿Te gustaría que un Sistema Avanzado de Asistencia al Conductor te avisara de esta situación en caso de no estar atento a la carretera?



Sí	28	93.3%
No	2	6.7%

En caso positivo ¿Cómo?



Sonido de intensidad dependiente del nivel de peligro	10	35.7%
Luz en el cuadro del coche	0	0%
Sonido y luz	16	57.1%
Otro	2	7.1%